Commodore
COMPUTER
CLUB
62
L. 5.000
La rivista degli utenti di sistemi Commodore
Anno VIII - N. 62 - 25 marzo 1989
Sped. Abb. Post. Gr. III/70 CR - Distr.: MePe
UN PROGRAMMA ANTI-VIRUS
Metti "Sheriff" nel tuo drive
MUSICA
Amiga in concerto
ASTRONAUTICA:
lancia il tuo satellite col C/64
C64, un pieno di assembly
Stampante, tutto quello che devi sapere
CAMPUS
LA TUA UNIVERSITA'
DEL SOFTWARE
systems

RINGRAZIA

LE AZIENDE

KODAK • BYK GULDEN EMOFORM • ALIVAR MOTTA • ENEL • RCA COLUMBIA PICTURES • SNAM GAS METANO • IL TELEGRAFO • NABISCO SAIWA CIPSTER • TRAINI E TORRESI TRASPORTI • MINOLTA COPIATRICI • CORSICA FERRIES - SARDINIA FERRIES • SIMCO DURANGO • CORRIERE DI PORDENONE • ISA - SPRAY PAN • PEG PEREGO • MIELE ELETTRODOMESTICI • FRANCESCO CONTI • WEIGHT WATCHERS • DE .CA. EDIZIONI • SCIARE • SYSTEM EDITORIALE • EDIZIONI UNIVERSO • FININVEST COMUNICAZIONI • SODALCO NUVENIA POCKET • DOLMA RAIDER • RACHELLI PASTICCERIA • PRIMUS MODA • DOCK MASTER • VMG JONATHAN • NTC THAT'S • DAVOLI LATTICINI • MINISTERO DELLA SANITÁ PER AIDS • DIOMEDE • VICK INT. OIL OF OLAZ • TONNO NOSTROMO • EZIO FIORI PUMA • BLOOMING CASUAL • OCCHIALI DILLIMAN • NUOVA POLTI VAPORELLA • POLENGHI LOMBARDO • ALSO ENERVIT • OSAMA SCRITTURA • RVR ELETTRONICA • IMPERIAL TV COLOR • AMSTRAD • ONOS HOLOWATCH • F.&P. RISO GALLO • RCS • IL GIORNO SAGISA

LE AGENZIE

YOUNG & RUBICAM • GRUPPO ESSEVI • TED BATES • RSCG • SAATCHI & SAATCHI • BTB • SPRING • CLIO ADVERTISING • M.F. MEDIA • PMS • MARKINGEGNO • KAPPADVERTISING • MARIO BELLI ASSOCIATI • McCANN ERIKSON • BHT • PLURAL • GRUPPO ODG • MEDIA TEAM • VEPROMARK • RIGHTMIX • TONIC • GRUPPO ETHOS • ELLECIZETA

PER AVER CREDUTO AL CIRCUITO
PIANIFICANDO LE PROPRIE CAMPAGNE PUBBLICITARIE NEL 1988/1989

CERTIFICAZIONE USCITE:

tramite il servizio della AGB Italia, Egimedia fornisce agli utenti di TOP ITALIA RADIO l'ufficiale certificazione delle uscite dei comunicati pubblicitari.

TARGET MIRATI:

il network TOP ITALIA RADIO, grazie al tipo di palinsesto unificato per 6 ore al giorno, dà la possibilità di posizionare le uscite su target specifici.

COPERTURA NAZIONALE:

grazie all'adesione di circa 200 radio provincia, TOP ITALIA RADIO INTEGRATO garantisce una omogenea illuminazione del territorio nazionale con ottime quote di penetrazione.

FLESSIBILITÁ GESTIONALE:

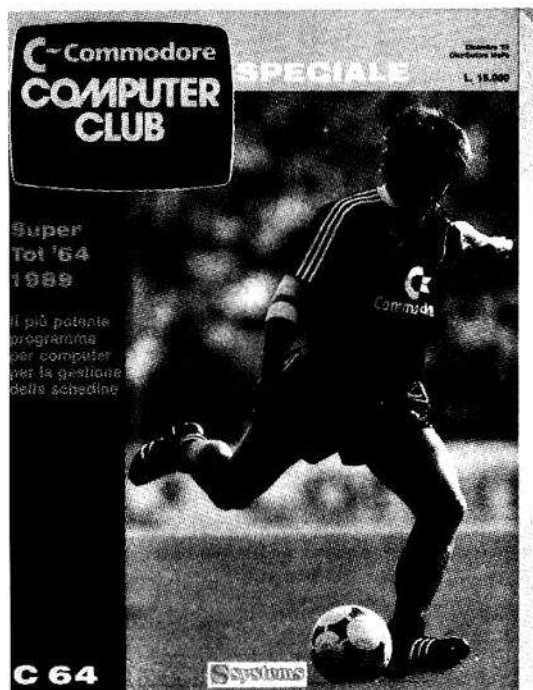
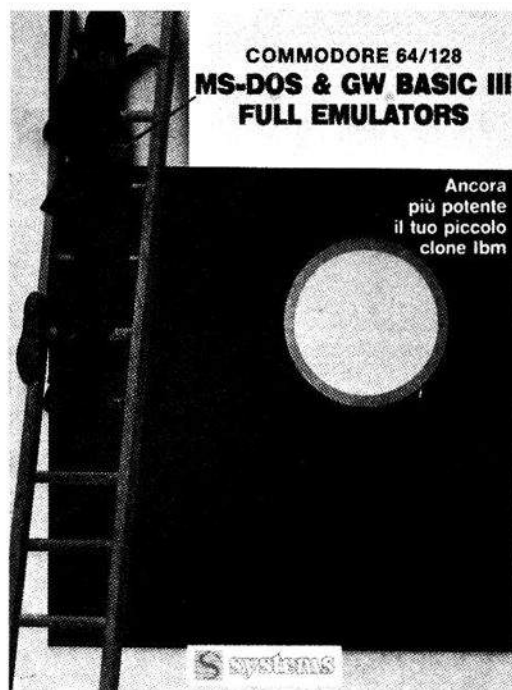
data la centralizzata produzione ed edizione del network, Egimedia offre le più ampie disponibilità a risolvere esigenze artistico-creative, tecniche, logistiche e commerciali ad ogni utente pubblicitario.

TOP ITALIA RADIO METTE COLORE E FANTASIA NELLA TUA VITA

EGIMEDIA s.r.l. - 20121 Milano - via della Spiga 1 - Tel. 02/798531 - 794592 - 798632 - Fax 780400

Sommario

RUBRICHE

Direttore: Alessandro de Simone - **Caporedattore:** Michele Maggi
Redazione/collaboratori: Paolo Agostini, Davide Ardizzone, Claudio Baiocchi, Angelo Bianchi, Luigi Callegari, Sergio Camici, Umberto Colapicchioni, Maurizio Dell'Abate, Valerio Ferri, Roberto Ferro, Cristina Magnaghi, Giancarlo Mariani, Roberto Marigo, Clizio Merli, Marco Mietta, Marco Miotti, Oscar Moccia, Roberto Morassi, Guido Pagani, Antonio Pastorelli, Sonja Scharrer, Fabio Sorgato, Valentino Spataro, Danilo Toma
Grafica: Arturo Ciaglia, Elena Salvadori
Direzione, redazione, pubblicità: via Mosè, 18 - 20090 Opera (MI) - Tel. 02/55500310
Pubblicità: Milano: Leandro Nencioni (direttore vendite), Angelo Ricupero - Via Mosè, 18 - 20090 Opera (MI) - Tel. 02/55500310
• Emilia Romagna: Spazio E - P.zza Roosevelt, 4 - 40123 Bologna - Tel. 051/236979
• Toscana, Marche, Umbria: Mercurio srl - via Rodari, 9 - San Giovanni Valdarno (Ar) - Tel. 055/947444
• Lazio, Campania: Spazio Nuovo - via P. Foscari, 70 - 00139 Roma - Tel. 06/8109679
Segreteria: Tiziana Sodano - **Abbonamenti:** Liliana Spina
Tariffe: prezzo per copia L. 5.000. Abbonamento annuo (11 fascicoli) L. 50.000. Estero: il doppio.
Abbonamento cumulativo alle riviste Computer e Commodore Computer Club L. 90.000.
I versamenti vanno indirizzati a: Systems Editoriale Srl mediante assegno bancario o utilizzando il c/c postale n. 37952207
Composizione: Systems Editoriale Srl - **Fotolito:** Systems Editoriale Srl
Stampa: Systems Editoriale/La Litografica Srl - Busto Arsizio (Va)
Registrazioni: Tribunale di Milano n. 370 del 2/10/82 - Direttore Responsabile: Michele Di Pisa
Sped. in abb. post. gr. III - Pubblicità inferiore al 70%
Distrib.: MePe - via G. Carcano, 32 - Milano
Periodici Systems: Banca Oggi - Commodore Club (disco) - Commodore Computer Club - Commodore Computer Club (disco produzione tedesca) - Computer - Computer disco - Electronic Mass Media Age -Energy Manager - Hospital Management - MondoRicambi - Nursing '90 - PC Programm (disco) - Personal Computer - Security - Software Club (cassetta ed. italiana) - VR Videoregistrare

CARO EDITORE, PROTESTO

Non sempre una Redazione è colpevole, anche se non sembra innocente

Quando un'azienda cambia sede è inevitabile che, nel periodo di tempo necessario per effettuare "materialmente" il trasferimento, si verifichino disguidi e ritardi; analoghi disguidi possono verificarsi quando un'azienda si espande: vecchi incarichi vengono affidati a nuovi operatori che prendono il posto di altre persone, passate ad altre mansioni.

Nonostante tali scusanti, senz'altro valide, vi sono dei "minimi" di efficienza che non dovrebbero essere superati. In particolare, di recente, si sono verificati spiacevoli(ssimi) equivoci che, in un'organizzazione più razionale, si sarebbero potuti evitare.

Ci riferiamo ai fascicoli speciali, con firma Commodore Computer Club, che avrebbero dovuto rappresentare, nelle intenzioni originarie, la risposta della Systems Editoriale alla massiccia vendita di C/64 e di drive Commodore 1541 riscontrata nel trascorso periodo natalizio. Una marea di telefonate e di lettere di nuovi utenti, infatti, ci ha spinto, fin dalle prime settimane dell'89, a riproporre in edicola alcuni prodotti di notevole successo.

Purtroppo le copertine degli stessi "speciali", succedutisi, per di più, secondo cadenze diverse dal quelle prefissate, hanno tratto involontariamente in inganno coloro che neofiti non sono e che erano già in possesso dei prodotti citati.

E' così successo che lo "Speciale Drive" ha visto la luce prima che fosse possibile presentarlo nelle nostre pagine (ed una presentazione tardiva non ha certo senso); "Totocalcio", "Emulatore Gw-Basic" e "La Voce III" sono invece apparsi in edicola privi dell'esplicita etichetta "ristampa" che avrebbe dissipato ogni dubbio nei nostri affezionati lettori. Solo l' "Emulatore Turbo Pascal 64", ma in ritardo ripetto alla tabella di marcia, ha visto la luce nei modi stabiliti.

La responsabilità dell'operazione commerciale, "ovviamente", viene frettolosamente attribuita alla Redazione (ed alla Direzione in particolare) che, in effetti, non c'entra per nulla, non essendo stata interpellata, nè in seguito informata, sull'aspetto finale delle copertine dei prodotti citati, mandati in stampa tutti insieme ed inviati anzitempo alla distribuzione.

Quello che ci preme sottolineare, e spero che l'editore non ce ne voglia, è che alcuni compiti vanno affidati a chi conosce il mercato; nel caso particolare a chi conosce i lettori, vecchi e nuovi che siano.

Se, in seguito alle proteste (dei lettori e nostre) qualcuno verrà destinato ad altre mansioni, ce ne dispiace, ma non più di tanto.

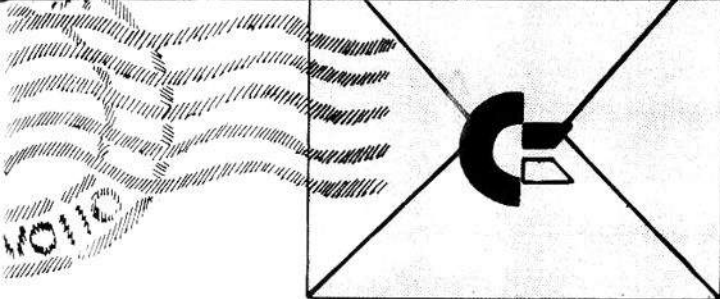
L'importante è che, chiariti gli equivoci, il lettore sia tenuto doverosamente nella giusta considerazione, magari informando, eventuali nuovi "esperti" rimasti nei ranghi, che una rivista di informatica va fatta in un certo modo; inserendo, per esempio, una sola puntata alla volta per ciascuna rubrica. E non tre, come è capitato di recente.

Alessandro de Simone

Caro Alessandro,

non ho nulla da eccepire alle tue critiche, che ritengo giuste e motivate. Ti prego di chiedere scusa ai lettori, da parte mia, ma di sottolineare, nel contempo, che le disfunzioni segnalate sono dovute ad "incidenti di percorso", del tutto fortuiti e, in ogni caso, involontari; e che, ovviamente, non verranno mai più ripetute.

Michele Di Pisa

(Editore Systems Editoriale)

la vostra posta

NE ABBIAMO GIA' PARLATO

Molti lettori (tra cui *Mario Simonetti di Caivano, Massimo Frisoni di Rimini, Giuseppe Sciacca, Alessandro Zarrilli di Poggibonsi, Fabrizio Cascioli di Roma, Vittorio Cascio di Enna, Luca Vignale di Brandizzo, Maurizio Piraccini di Cesena, Romano Demichelis di Milano, Ciro Nigri di Verona, Davide Amato di Trapani, Emilio Cherubini di Verona, Vittorio Binassi di Correggio, Francesco Russo di Cerignola, Marco Rocconi, Francesco Varone di Bellona, Sergio Piras di Cagliari, Corrado Giorgini di Bolzano, Eugenio Tacchini di Rivergaro, Marco Frazzioli di Montecatini, Antonio Russo, Michele Cipollone di Orsogna, Marco Catullo di Frosinone, Maurizio Cibello di Salerno, Ivan Esposito di P.ra Borromeo, Alessio Isaja di Trieste, Marco Simonetti di Caivano, Rocco Manuele di Scordia, Daniele Zaganella, Claudio Mele di Milano, Angelo Barroco di Asti, Pietro Grilli di Bolzano, Maurizio Cecchinetti di Bologna, Luciano Guidi di Chieri, Alessandro Feliziani di Osimo, Sante Ferrara di Fasano*) chiedono argomenti affrontati più di una volta su C.C.C. o che saranno, tra breve, trattati. Alcuni pongono, inoltre, domande la cui risposta presuppone una conoscenza più che approfondita del Basic o, addirittura, del Linguaggio Macchina. Pur apprezzando il giusto desiderio di saperne di più, non possiamo far altro che sottolineare l'importanza di una documentazione vasta e precisa. Consigliamo, pertanto, di leggere con la massima attenzione i libretti di istruzione dei computer e delle periferiche che possedete, i depliant su cui sono riportate le caratteristiche delle macchine e l'attento studio dei fascicoli di Commodore Computer Club collezionati nel frattempo. A chi ci segue da pochissimi numeri, invece, suggeriamo di considerare la nostra rivista come un'enciclopedia a dispense, da conservare gelosamente e da consultare come un vero e proprio manuale. La raccolta dei fascicoli arretrati, quindi, può rappresentare una valida opportunità per coloro che intendano dedicarsi, seriamente, allo studio del personal computing.

KICKSTART ED EPROM

☐ **Le due versioni del Kickstart di Amiga (1.2 ed 1.3) sono compatibili tra loro? Che cosa è una memoria Eprom?**
(Massimiliano Papadia - Foggia)
(Ciro Nigri - Verona)

• Anche a noi è giunta voce (o meglio, un pettegolezzo) sulla difficoltà di far funzionare, con Kc 1.3, programmi che giravano perfettamente "sotto" Kc 1.2. Finora non ci è capitato di sperimentare tali incovenienti e, anzi, posso assicurarti di aver visto girare, sul nostro recentissimo A.2000 (dotato di Janus AT, di 68020, coprocessore 68882, di Kc 1.3 e di 4 mega di Ram) programmi in Assembly da noi sviluppati con Kc 1.2 "supportati", ovviamente, sia da Wb 1.2 che 1.3.

E' probabile che la colpa sia delle software house che, nello scrivere i vari programmi (poi commercializzati), non hanno tenuto conto degli "entry point" suggeriti dalla stessa Commodore. In questi casi, quindi, la modifica delle Rom può creare incompatibilità. Non appena ne sapremo di più, come al solito, ne riparleremo con maggior cognizione di causa.

Una EPROM (Erasable Programmable Read Only Memory = Memoria cancellabile programmabile a sola lettura) è un circuito integrato caratterizzato da una finestrella trasparente posta alla sua sommità. Si tratta di una memoria permanente che, una volta programmata mediante un apposito apparecchio (denominato, appunto, "Programmatore di Eprom"), consente di tenere memorizzato il programma o i dati come se fosse una comunissima Rom. La differenza consiste nel fatto che mentre una Rom non può esser cancellata, una Eprom può essere riprogrammata dopo averla sottoposta, per un periodo di tempo opportuno, ad un'intensa fonte di luce ultravioletta (ecco spiegato il motivo della finestrella). Per evitare che la luce cada sulla Eprom, è buona norma coprirla con una etichetta autoadesiva opaca.

RS-232 PER C/64

☐ **Sul N. 49 di C.C.C. è riportato un articolo ("Dialogo tra computer") in cui viene descritta la realizzazione di una interfaccia Rs-232 per collegare un C/64 con un computer Ms-Dos compatibile. Manca, però, l'indicazione del condensatore C-5.**
(Giuseppe Sciacca)
(Nino Ferdeghini - La Spezia)

• L'interfaccia descritta, che può mettere in contatto il C/64 (il C/128 e, addirittura, il Vic/20!) con una qualsiasi apparecchiatura dotata di Rs-232 (Ibm compatibili, Amiga, stampanti e così via) non è altro che un semplicissimo apparecchio costituito da un paio di integrati e qualche altro componente passivo. Il condensatore citato non svolgeva altro compito se non quello di "livellare" la tensione. Nel progetto, infatti, il componente era stato previsto ma, in seguito, abbiamo notato che il prototipo funzionava egregiamente anche senza, dal momento che la porta utente del computer dispone già di tensione più che soddisfacente. Se proprio ci tieni, comunque, non devi far altro che collegare il condensatore in oggetto in parallelo alla linea di alimentazione.

AMIGA E HARD DISK

☐ **E' possibile usare un qualsiasi Hard Disk sull'Amiga oppure è necessario usarne uno originale Commodore?**
(Anonimo appassionato)

• In teoria tutto è possibile ma, spesso, in pratica...

Non penso che, tuttavia, sia necessario ricorrere ad accessori non originali dal momento che i prezzi di vendita sono sostanzialmente identici.

PULSANTE DISPETTOSO

☐ **Ho costruito un circuitino (di cui allego schema) che, basato su un deviatore collegato al pin di Reset, dovrebbe svolgere un certo compito. Mi accorgo, però, che azionando il deviatore (senza premere il pulsante di reset) il computer si resetta egualmente. Come mai?**
(Simone Chifari - Cesena)

• Il circuito stampato di un computer è attraversato da una tensione bassissima, ma da segnali che viaggiano ad una frequenza (imposta dal quarzo) piuttosto elevata. In una situazione così delicata è sufficiente una minima variazione per azionare, involontariamente, alcune procedure hardware non richieste.

Quando sposti il pulsante di un deviatore, infatti, modifichi, anche se di poco, una capacità, ed esattamente quella presente tra i due cavetti che collegano il deviatore stesso alla piastra. E' probabile che questa alterazione, benché minima, provochi la generazione di un segnale che viene "in-

AIUTATECI A SERVIRVI MEGLIO

Spesso alcuni lettori, che dichiarano di possedere numeri arretrati del nostro periodico, porgono quesiti le cui risposte sono già state esplicitamente pubblicate (in occasione di risposte ad analoghe domande) oppure sono contenute in articoli presenti nei fascicoli in loro possesso.

Per evitare di ripetere argomenti già trattati, pertanto, ricordate di indicare sempre, nelle lettere che inviate, i numeri dei fascicoli in vostro possesso: potremmo infatti indicarvi, se esistono, gli articoli che, in un modo o nell'altro, chiariscono gli argomenti richiesti.

Si ricorda ai lettori che non ci è possibile rispondere privatamente, nemmeno se si acclude l'affrancatura per la risposta.

Per accelerare il servizio, ricordate di indicare sempre la data di spedizione dal momento che questa costituisce diritto di precedenza.

A tutt'oggi (con la pubblicazione del presente fascicolo) risulta evasa l'intera corrispondenza pervenuta in Redazione entro il 16 febbraio.

Coloro che non vedessero citato il loro nome nella presente rubrica, possono trovare risposta ai loro quesiti in appositi articoli, apparsi nel frattempo, o di prossima pubblicazione.

terpretata" come un collegamento a massa del pin, con le conseguenza che hai notato.

Ti posso assicurare che è sufficiente una minima svista per provocare guai incredibili e di difficile individuazione. Alcuni computer, infatti, devono la loro affidabilità alla perfetta realizzazione del circuito stampato che non presenta inconvenienti di sorta tra piste adiacenti.

Un problema analogo a quello da te notato sorge nel caso si desideri realizzare un doppio pulsante di reset, uno per il drive ed uno per il computer. Anche in questo caso sembrerebbe sufficiente una porta And e due pulsanti ma la stessa lunghezza dei due cavetti può creare "interferenze" che provocano reset involontari.

SUPERCOPIATORI

□ In molti dei vostri articoli affermate che non esiste un sistema di protezione sicuro perchè, ormai, sono in commercio numerose cartucce in grado di aggirare qualsiasi protezione. Quali sono?
(Maurizio Ascione - Ercolano)

• E' sufficiente recarsi presso il più vicino negozio realmente specializzato, per scegliere con calma tra le varie dozzine di cartucce in vendita. Ormai quasi ognuna di queste contiene, tra le varie opzioni, una di copia "totale" (da nastro/disco verso nastro/disco). Il sistema, di solito, è semplicissimo e si basa sulla copia dell'intero contenuto della memoria del computer sul quale sta girando il programma da duplicare. L'unico inconveniente è rappresentato dal fatto che i file copiati sono quasi sempre un po'... lunghi dal momento che il copiatore non può stabilire se esiste una zona Ram in cui sono presenti dati insignificanti: copia tutto ciò che può!

Molto spesso abbiamo parlato di cartucce specifiche e non ci sembra il caso di parlarne ancora; a meno che non ci capiti tra le mani qualche novità in tal senso.

ADATTATORE 6499

□ Non riesco a far funzionare correttamente il mio adattatore telematico 6499.
(Numerosi lettori)

• Abbiamo più volte pubblicato lamentele in tal senso; in certi casi la colpa è della Sip, o meglio della sua "rete" che sembra risalire ai tempi dei Borboni; a volte, invece, la causa è da ricercarsi nel libretto di istruzioni, carente ed inadeguato sotto molti punti di vista.

In entrambi i casi cominciamo ad essere un po' stufi di vederci addebitate colpe non nostre.

E chi ha orecchie per intendere, intenda (o almeno se le lavi più spesso: può essere un buon inizio).

LE SORELLE AMIGA

□ Quale è la reale differenza tra un A-500 (espanso a 1 mega) ed un A-2000 inespanso?
(Silvia e Riccardo)

• Le due "sorelle" sono destinate ad utenti un po' diversi tra loro. Chi non vuol spendere cifre da capogiro (un A-2000 completo può costare oltre sette milioni!) ma preferisce espandere dignitosamente il computer, può "fermarsi" all'Amiga 500. Chi, al contrario, ritiene che la compatibilità Ms-Dos sia vitale o che l'espandibilità a 8 mega sia una necessità, prima o poi, inderogabile, allora dovrebbe iniziare con un A-2000.

La compatibilità tra le due macchine configurate "alla pari", quindi, è totale e completa. Rimane il fatto che ad un A-500 non è possibile collegare più di due disk drive 3.5 esterni (con A-2000 è possibile collegarne tre da 3.5 oltre a quello da 5.25) e non per questioni di incompatibilità, ma di alimentazione: l'alimentatore di A-2000 è più potente proprio perchè è prevista la connessione di numerosissimi accessori sia negli slot interni sia esternamente.

Ripetiamo ancora una volta, tuttavia, che un Amiga DEVE esser dotato di almeno due drive e di 1 mega byte di Ram. In caso contrario si lavora con una certa difficoltà.

TROPPO POTENTE

□ Il mio C/64 assicura che 3 elevato alla quarta potenza è eguale a 81.0000001. Io credevo di meno...
(Vittorio Cascio - Enna)

LA VOCE DEI LETTORI

OPINIONI

La rubrica "La voce dei lettori" è nata per dare spazio (e sfogo...) ai lettori che, pur non avendo nulla di particolare da chiedere, intendono far sentire la propria voce, chiedere solidarietà ad altri utenti o, semplicemente, riportare osservazioni ritenute utili o importanti. Per questioni di spazio le lettere sono spesso condensate; vengono poi "censurate" le osservazioni ed i termini troppo "forti" che possano arrecare offesa ai lettori, agli inserzionisti e... a noi stessi. In calce viene riportata la sola iniziale del cognome del lettore (salvo casi particolari), per fare in modo che questi possa sentirsi ancor più libero di esprimere la propria opinione. Coloro che desiderano il minor numero di "tagli" possibile alle loro missive sono pregati di inviarle su DISCO, usando il word processor Easy Script.

AMIGASHOP

UNICA SEDE: VIA MAC MAHON, 75 - 20155 MILANO

Tel. 02/323492 solo per negozio e informazioni relative acquisti in Milano - direttamente in sede
Tel. 02/33000036 per ordinazioni da tutta Italia
Fax 02/33000035 in funzione 24 ore su 24
BBS MODEM 02/3270226 (banca dati) al pomeriggio dopo le 13.00 fino al mattino successivo.
Aperto al pubblico nei giorni feriali dalle 9.00 alle 12.30 e dalle 15.00 alle 19.00
e il sabato dalle 9.30 alle 13.00 e dalle 14.30 alle 18.30 - chiuso il lunedì

MK5 PROFESSIONAL ACTION REPLAY

ORA AVETE FATTO "13"

1) MK 5, la migliore per effettuare copie di sicurezza;
2) il velocizzatore più sicuro e compatibile tape o disc;
3) microprocessore studiato per sostituirsi a quello del computer, invisibile e trasparente al sistema stesso;
4) ricerca le poche poke per donare vite infinite a tutti i programmi gioco;
5) possibilità di avere giochi in versione trainer senza nozioni di linguaggio macchina;
6) sprotegge da nastro a disco ed all'inverso qualsiasi programma anche in multiload;
7) trasferisce da 5 1/4 a 3 1/2;
8) velocizza 5/10 volte il nastro;
9) rende parallelo il disc drive (202 blocchi in 6 secondi)
10) editor di schermo per cambiare scritte nei programmi;
11) interfaccia parallela centronics standard grafica (con apposito cavo opzionale);
12) salva e stampa schermate e sprite di qualsiasi gioco, con possibilità di alterazione personalizzata;
13) nuovi comandi basic, monitor di linguaggio macchina, immagini su nastro in successione e tanto di più!!

ECCO PERCHE' HAI FATTO "13"
Per Commodore 64 e 128, compatibile con tutti i registratori e tutti i drives in modo 64.
E per fare "13" vi occorrono solo
L. 105.000 (IVA compresa)

TELEVIDEO 64

Novità assoluta ora puoi ricevere il televideo Rai direttamente dal tuo Commodore 64/128 con possibilità di stampa: è sufficiente possedere un qualsiasi tv con presa scart o monitor. Manuali in italiano.
L. 149.000

PAGEFOX

La più potente cartuccia per desk top publishing grafica di stampa professionale, supera addirittura Geos e le sue applicazioni, disegno con vari tipi pennello scrive con scelta tipi di carattere.
Manuale in italiano
L. 119.000

SPEEDOS PLUS "NEWEL"

Il miglior tipo di velocizzatore parallelo per 64/128 (modo 64) inimitabile ed affidabile al 100%, compatibile con tutto il software commerciale disponibile in tutte le versioni: 64 e 1541; 64 e 1541/II; 64 e OC118; 64 e 1541c; 64 nuovo modello e drive; C128 e 1541.
Tutti i tipi modello super a
da L. 59.000

REALTIME (provideo) 64

Digitalizzatore 16 tonalità di grigi in tempo reale (fotografico) per ottenere immagini senza fermo immagine di qualità eccellente.
L. 149.000

PROGRAMMATORE EPROM

Per programmare fino ad Eprom 256 K di qualsiasi provenienza, con il tuo piccolo 64 potrai copiare, verificare, comparare Eprom di qualsiasi computer o apparecchiatura.
L. 229.000

MODEM DIRETTO NEW

Modem dedicato 300 baud co software e cavi collegamento per C64.
L. 69.000

INTERFACCIA MIDI 64 - AMIGA

Potrai finalmente collegare tastiera Midi al tuo Amiga C64 (per tutti i musicisti ed hobbisti musica).
L. 79.000

THE CARTRIGE

Se non vuoi spendere una cifra enorme e ti accontenti di fare tutto ciò che fa Mikie, con turbo, sprite killer, poke, monitor, tool kit, più un eccezionale sprotettore cassetta disco e viceversa in un unico file, completo di manuale in italiano a sole.
in offerta L. 69.000

MOTHER BOARD 64

Per collegare al tuo 64/128 da 1 a 3 cartucce, anche contemmporanee, senza dover spegnere la tastiera con reset.
da L. 55.000

PROCESSORE VOCALE

L'unico digitalizzatore sonoro e musicale, campionatore di suoni e parole, con possibilità di dare comandi voce al computer, completo di microfono software in italiano e manuali in italiano (Made in Newel)
L. 119.000

FINAL CARTRIGE III ORIGINAL

La più bella cartuccia ad icone come the cartrige, con dump video, orologio agenda, e sprotettore. Dai il fascino dell'Amiga al tuo C 64.
L. 79.000

PRO SOUND DESIGNER

Ovvero Elaboratore professionale del suono. E' un campionatore sonoro che funziona su tutti gli Amiga, 8 bit stereo sampler da 1 a 28 KHz mono e da 1 a 17 KHz stereo; playback a 35 KHz, avanzate funzioni di editing e compatibile anche con altri pachhetti software come ad esempio: Sound sampler, Future sound, Perfect sound, ecc;

L. 179.000

MINI GEN

MINI-GEN una grande novità per professionisti ed entusiasti, per ottenere sovrapposizioni di animazioni, titoli, messaggi ecc. Funziona con tutti gli Amiga ed è compatibile con programmi come TV-Text, Pro video e molti altri. Ora la videotitolazione è alla portata di tutti, semplicissimo da usare.

L. 399.000

NOVITA' PER I POSSESSORI DI AMIGA

Vuoi finalmente lavorare senza il fliker eccessivo del tuo "Amigo". Multisync Nec II 14" Color. In offerta limitata a:

L. 1.300.000

(Iva e trasporto completo)

VIDEON

Basta con i noiosi filtri per i vari passaggi... Ora c'è VIDEON! Il VIDEON è un digitalizzatore video a colori dotato di un convertitore PAL-RGB con una banda passante di 15 KHz per ottenere immagini a colori dalle stupefacenti qualità... Funziona in risoluzioni di: 320x256 - 320x512 - 640x256 - 640x512. Può essere collegato a una qualsiasi fonte video PAL, ad esempio videoregistratori, computer, telecamere, televisori, ecc. Il prodotto permette di visualizzare il segnale video collegato all'apparecchio e in più permette la regolazione di luminosità, colore, saturazione, contrasto. E' corredato di software che permette la manipolazione di immagini IFF HOLD MODIFY da 32 a 4096 colori con tecniche di SURFACE-MAPPING su solidi geometrici. E' in arrivo la versione 2.0

L. 420.000

AMIGA CARD

Hard disk in AmigaDOS per l'Amiga 2000 su scheda, semplice da installare e lascia libero lo spazio per il secondo drive interno. Disponibili anche versioni esterne per A500 e A1000. Disponibile anche versione con AUTOBOOT

L. 990.000

BOOTSELECTOR

Utilissimo per usare DF1: come se fosse il DF0: senza perdere memoria.

L. 23.000

ANTIRAM

Permette di scollegare una qualsiasi espansione Amiga (senza saldature)

L. 23.000

DISKCONNECT

Per sconnettere il secondo drive del Amiga e recuperare memoria

L. 23.000

TASTIERA MIDI

Musicale, Amiga compatibile, Pro Sound designer, Midi

NEW

ESPANSIONI

512K originali Commodore per A500

L. 319.000

Disponibili espansioni di memoria per A500, 1000, 2000 interne ed esterne da 512K fino ad 8MB. Telefonare per ulteriori informazioni.

L. telefonare

KICKSTART 1.3 ROM

Il nuovo sistema operativo dell'Amiga ora in ROM applicabile facilmente su A500 a A2000 senza saldature e senza perdere il vecchio s/o 1.2 VARIE VERSIONI!!

L. 119.000

2MB esterne autoconfiguranti profex per A500 **L. 1.299.000**

AMIGA MODEM 2400 PAK

Modem dedicato per A500 - A1000 - A2000, esterno 300, 1200, 2400 baud (V21-22-22BIS). Autodial, autoanswer, Hayes compatibile, completo di software e cavo di connessione al computer (disponibili altre versioni 300/1200 e 300/1200-1200/75 Videotel)

L. 399.000

i nostri disk drive

disk drive slim, MECCANICA NEC BEIGE

sono disponibili:

per Amiga 500 3,5 pollici passante **L. 229.000**

per Amiga 500 5,25 pollici 40/80 tracce passante **L. 329.000**

per Amiga 2000/interno **L. 179.000**

per C-64 OCC118 **L. 249.000**

Disponibile finalmente AmigaFax (NOVITA') L. 199.000

6ACCESSORI PER L'AMIGA

EASY SOUND	L. 119.000	Digitalizzatore Audio IFF compatibile Sonix ecc.
EASY VIEW	L. 119.000	Digitalizzatore Video compatibile Digiview
DIGI AUDIO & VIDEO	L. 189.000	Digitalizzatore Audio & Video: tutto in uno come sopra
SERIAL AMIGA	L. 39.000	Interfaccia per collegare stampanti seriali/64 all'Amiga
INT. MIDI PROF.	L. 79.000	Per collegare tastiera MIDI all'Amiga (Passthrought)
PAL GENLOCK	L. 590.000	Genlock amatoriale con regolazioni per A500, A1000, A2000

PORTADISCHI 40pz.	L. 20.000	KIT PULIZIA 3,5"	L. 10.000
PORTADISCHI 60pz.	L. 30.000	KIT PULIZIA 5,25"	L. 10.000
		PORTADISCHI POSSO	L. 35.000 (100 posti)

Scheda XT Janus (compatibilità 100% MS DOS) **e AT Janus per Amiga 2000**

L. telefonare

Sono disponibili i programmi di Fish, di Public Domain e relativo manuale d'uso in italiano. **L. 2000 il catalogo**

HARD DISK CARD

per A2000 in modo MS-DOS (meccanica Miniscribe, Controller Westerndigital)

20 MB	L. 639.000
33 MB	L. 799.000
40MB	L. 969.000
20 MB HARD DISK	L. 539.000

SERVIZI OFFERTI

Per questioni di spazio non ci è possibile elencare moltissimi altri articoli. Veniteci a trovare o richiedeteci i cataloghi settoriali. Chi verrà, a trovarci con questa rivista e acquisterà almeno L. 100.000 (centomila) di prodotti accessoristici hardware e software riceverà, richiedendolo, un "controvalore" pari al costo della rivista.

Tutto il materiale è garantito 12 mesi + 7 giorni di prova, ed è in pronta consegna

Spedizioni in contrassegno postale in tutta Italia in REALTIME (servizio computerizzato) 02/33000036 (da martedì a venerdì dalle ore 9.15 alle 18.50)
Servirsi per ordini esclusivamente dei numeri indicati o del Fax

Ricorda alla Newel trovi anche tutto per C-64/128, Amiga, Atari, PC Amstrad ecc. Richiedi il catalogo specificando il computer posseduto.
Richiedi il nostro nuovo catalogo per Amiga con tutte le ultime novità hardware & software, oltre 1000 programmi selezionati... inviando L. 2.000 in francobolli.

CHIAVI IN MANO – PREZZI IVA INCLUSA – TUTTO COMPRESO

• Ed hai perfettamente ragione se, invece di calcolare una potenza, fai calcolare un prodotto; cioè:

```
Print 3 * 3 * 3 * 3
81
READY.
```

Vi sono, infatti, numerosi algoritmi (= procedimenti logici) per calcolare una potenza. Gli uni differiscono dagli altri sia per la precisione ottenibile sia per la lunghezza del programma necessario a risolvere il problema.

E' inutile dire che, nel caso del C/64 (e dei piccoli computer domestici in generale) i progettisti sono stati costretti a fare i conti con la memoria disponibile. Ne è risultato un compromesso che se può essere accettabile nei casi più complicati (elevazione di valori decimali a potenze rappresentate da numeri non interi) porta a risultati imbarazzanti nei casi più banali, come quello da te evidenziato.

Fortunatamente, però, è possibile aggirare l'ostacolo facendo in modo di elaborare (quando possibile) non una potenza, ma un prodotto: una potenza, non dimentichiamolo, non è altro che una "ripetizione" di prodotti.

HARD COPY DEL VIDEO

□ Come posso trasferire su carta la schermata di una pagina di testo visualizzata in un programma professionale?
(Cristoforo Paiano - Lecce)

• I moderni computer professionali (tra cui gli Ms-Dos compatibili) dispongono di un tasto che consente di trasferire su carta ciò che compare sul video in un particolare momento. Purtroppo ciò non è possibile farlo con un C/64 perchè, a livello hardware, il piccolo computer non è predisposto per tale operazione.

Sono commercializzate, però, alcune cartucce che consentono, mediante la pressione di un pulsante, di riportare su carta sia la schermata hi-res sia quella low-res.

E' infatti necessario che sia presente, nella memoria del computer, sia la routine idonea alla stampa, sia la possibilità di attivarla mediante la pressione di un certo tasto.

Quando si fa partire un programma professionale (oppure un listato di cui non si conosce l'esatto funzionamento) non è possibile "dirottare" l'elaborazione verso la nostra routine, che quindi, pur se presente, non potrà mai funzionare.

Se, invece, è possibile modificare il programma, sarà uno scherzo aggiungere un paio di righe per fare in modo di ottenere la copia su carta.

Se, ad esempio, il programma principale è in Basic e la routine di h-copy anch'essa in Basic, la modifica da apportare può esser la seguente:

```
...get a$: if a$ = "s" then gosub 5680...
```

E' ovvio che la riga 5680 deve essere la prima della routine Basic che, lo ricordiamo, deve terminare con il comando Return.

Se, invece, la routine è in linguaggio macchina e la locazione di partenza è 49152, la modifica sarà del tipo...

```
...get a$: if a$ = "s" then sys 49152...
```

In entrambi i casi premendo il tasto "s" sarà eseguita la copia su carta.

LA VOCE DEI LETTORI

ABBANDONI

...ora anche la vostra rivista sta abbandonando noi 128-isti; avevate promesso addirittura articoli sul CP/M che non si sono visti. Perfino i piccoli programmi di utilità (orologi, calcolatrici) non vengono più pubblicati. Non crediate con questo di incrementare le vendite di Amiga perchè i numerosi 64-isti aspettano che il numero e la qualità dei programmi sia quantomeno eguale o superiore al C/64 perchè non basta una buona grafica ed una buona musica per fare bello un programma. PER CASO VOLETE FAR FARE AL C/128 LA FINE DEL VIC-20?
(Ludovico Q. - Cetraro)

MINI - ERRORE

□ La linea 740 del mio programma ("La data infinita" C.C.C. n. 60) deve terminare con Goto 750 e non Goto 75 come appare a pagina X dell'inserto.
(Mario Saggese - Benevento)

• Ci scusiamo con i lettori (e con l'autore) per l'errore di impaginazione segnalato: lo zero finale è stato "tagliato" via involontariamente.

Saggese, inoltre, ricorda che il programma in oggetto (privo delle modifiche indicate alla fine dell'articolo) è idoneo a stabilire una qualsiasi data compresa tra l'anno 0 e il 1699 e tra il 1901 e il 2099.

LA VOCE DEI LETTORI

FILOSOFWARE

Ho realizzato (quasi per scherzo) un breve programma che indica la condotta da seguire per esser felici, eliminando ciò che impedisce la realizzazione della felicità.

Riflettendo, ho pensato di sostituire all'elementare problema della felicità un qualunque problema filosofico. Infatti anche la ricerca della felicità è un problema filosofico (cfr. Platone, Epicurei e Stoici). L'idea che un computer possa risolvere questioni di filosofia mi lasciò perplesso ma ancor più sorpreso ed affascinato; infatti affidare tale compito ad un computer avrebbe l'immenso vantaggio di eliminare l'elemento umano di condizionamento inconscio ("Si è portati ad accettare più ciò che si desidera che ciò che ci mostra la ragione" - Nietzsche).
(Cristano Giuseppe M.)

TOTOCALCIO

Come già affermato la volta scorsa, lo speciale "Totocalcio" non è altro che una ristampa del precedente "speciale". Siamo stati spinti alla pubblicazione dalla notevole richiesta da parte dei nuovi utenti di C/64.

I programmi che compaiono sul dischetto allegato, quindi, sono identici a quelli già presenti sul nastro della precedente edizione.

Vi sono, pertanto, alcuni "errori" dovuti al fatto che il supporto magnetico, stavolta, è il disco e non il nastro.

Per far funzionare correttamente i vari programmi consigliamo di sostituire il dischetto originale con uno vuoto (già formattato) soprattutto al momento in cui compaiono domande del tipo "Vuoi registrare?". Il programma "Routine Correct"

LA VOCE DEI LETTORI

PROTESTE

Trovo diseducativo divulgare articoli su come realizzare i Virus. Ci sono fin troppi delinquenti che mettono in circolazione dischetti infetti e non è il caso di favorire ulteriormente il vandalismo informatico. Sarebbe più opportuno, invece, scrivere articoli su come neutralizzare i virus.

Per ciò che riguarda le foto di donne nude, non troppo castigate, apparse di recente su C.C.C. ritengo che siano fuori luogo per una rivista di informatica. Non è da dimenticare (anche se sono una minoranza) che C.C.C. viene letta anche dal pubblico femminile. Non avete pensato che potrebbero sentirsi offese e trascurate e quindi decidere di non comprare più la rivista? Come vi sentireste, voi, se aprendo la vostra rivista di INFORMATICA preferita vi trovaste davanti la foto di un uomo nudo?

Che cosa direste?

(Claudio F.)

"Wow!"

(lungo 14 blocchi) pone questa domanda dopo aver trasferito in memoria i codici in linguaggio macchina. Prima di premere il tasto Return, quindi, sostituite il dischetto. Per esempio, prima di far partire il programma "Routine Correct" con il solito Run, cambiate il nome del file presente nella riga 210 ("Routine correct") con un altro qualsiasi e diverso, in ogni caso, da altri nomi eventualmente già presenti sul dischetto stesso. Al termine della riga 210 ponete anche un comando End. Quest' ultimo accorgimento non è strettamente necessario, ma ricordiamo che, al termine della registrazione, dovete spegnere e riaccendere il computer.

Sottolineamo infine che alcuni programmi presentano incompatibilità con determinate cartucce o velocizzatori software dal momento che utilizzano identiche locazioni di memoria. In caso di malfunzionamento, quindi, prima di ripetere le varie operazioni, staccate la cartuccia, oppure rinunciate ad utilizzare le utility di caricamento rapido.

PRETENDER TROPPO

□ Ho esaminato alcuni "compilatori" Turbo Pascal per C/64, ma mi sembra che siano caratterizzati da una velocità di elaborazione molto modesta.
(Manuele C. - Genova)

• Dunque, vediamo di intenderci. Vi sono, in commercio, computer appartenenti alla fascia professionale (prezzo dal milione e mezzo in su) e home computer (dalle 500 mila in giù). Corrispondentemente vi sono programmi idonei ai primi (dalle 200 mila in su) e ai secondi (dalle 30000 in giù).

Ciò che si desidera (un po' troppo ingenuamente) è far svolgere, ad un piccolo computer, compiti adatti ai fratelli maggiori Ms-Dos. Abbiamo già detto centinaia di volte che un C/64 completo, dotato di opportuno s/w, può essere considerato come una dignitosa stazione di videoscrittura. Ciò non toglie, però, che non è possibile eseguire compiti che sono quelli tipici dei computer professionali. Una cosa è il w/p, insomma, una cosa è il DTP.

E' ovvio che se vogliamo limitarci a scrivere la sola nota della spesa, un C/64 ed un super extra 80386 da 20 milioni arrivano allo stesso identico risultato.

Come è possibile, insomma, pretendere che un Turbo Pascal per C/64 sia in grado di competere con quello originale della Borland, destinato a girare su computer enormemente più veloci e completi?

Lo scopo di questi linguaggi (o emulatori, tra cui il nostro) è semplicemente didattico: lo studente che, a scuola, opera con un computer Ms-Dos compatibile dotato di Turbo Pascal, non potrebbe esercitarsi a casa sua, dove lo aspetta un modesto C/64.

Il fatto, poi, che un certo listato "gira" sul C/64 in un tempo medio di 10 secondi, mentre il PC della scuola compie lo stesso lavoro in meno di un secondo, è di secondara importanza, sempre se consideriamo la faccenda da un punto di vista strettamente didattico.

Del resto non vi sarebbe motivo di produrre computer sempre più complessi (e costosi) se tutti i problemi potessero esser risolti con il C/64.

CASI PARTICOLARI

□ Con il mio C/128 ho notato alcune stranezze. Ad esempio, tenendo premuti contemporaneamente i tasti Crsr Right, Crsr Down, Shift (sinistro) e Commodore si verifica uno sfarfallio sul video.
(Umberto Vecchi - Gargallo)

• Ogni 60-mo di secondo viene automaticamente attivata una routine che ha il compito, tra l'altro, di esaminare se viene premuto qualche tasto e, in caso affermativo, prendere opportuni provvedimenti.

La routine di scansione della tastiera, ovviamente, tiene conto di un utilizzo "normale" della tastiera stessa e prende in considerazione la pressione di un singolo tasto, oppure la pressione contemporanea di più tasti, a patto che la particolare combinazione sia effettivamente prevista (maiuscole, colori, shift + Run/stop e così via).

In teoria, se la pressione di più tasti non corrisponde ad una combinazione "lecita", l'effetto conseguente dovrebbe esser

LA VOCE DEI LETTORI

ANTI 64-ISTA

Secondo me il C/64 è servito solo per giocare e non ditemi di no! Che cosa ha infatti il C/64 in più dei suoi fratelli C/16, Plus-4, C/128 e Amiga? E' stato progettato e costruito per giocare e non certo per studiare o imparare, visto che ha una schifezza di Basic, per non parlare del suo S.O. tra i più obsoleti in assoluto. E passiamo al GEOS: è un FINTO S.O. la cui lentezza non è paragonabile al più misero S.O. sempre che ne esista uno più schifoso. In conclusione, perchè nascondere una triste realtà? Il C/64 è un videogame; se si vuol fare sul serio, si usa il Plus-4. Ritengo che il C/64 abbia avuto successo solo perchè, per moda, è stato "imposto". Aveva ragione Dante quando scrisse:

"e ciò che fa la prima, e l'altre fanno,
addossandosi a lei, s'ella s'arresta,
semplici e quete, e lo 'mperchè non sanno;"
(Purgatorio, terzo canto)

(Nino P. Cagliari)

nullo, oppure identico a quello ottenibile con la pressione di un minor numero di tasti.

In pratica, però, possono verificarsi malfunzionamenti sia a causa di routine che, per casi particolarissimi, "dirottano" il percorso del programma verso altre direzioni, sia per inconvenienti di tipo h/w dovuti al fatto che il chip responsabile della decodifica della tastiera entra in tilt.

DRIVE 1581

□ E' possibile usare il drive 1581 da 3.5 pollici con il C/128 in modalità CP/M?
(Massimo Vailati - C.llo B.mo)

• Non sembra che sia possibile formattare un dischetto da 3.5 in modalità CP/M, perchè l'operazione sembra partire, ma non raggiunge mai un termine. Inoltre, per caricare il comando Format (presente sul dischetto 5.25 Commodore) è necessario procedere ad un'operazione un po' rischiosa: caricare Format con il 1541, sconnettere tale drive e sostituirlo con il 1581 prima di dare conferma. Il rischio di bruciare qualche integrato è quindi notevole ma, anche se tutto procede bene, la formattazione non si verifica (abbiamo già tentato noi stessi).

ACCENDERE LA STAMPANTE

□ Se il mio C/128 è collegato alla stampante (e questa non è accesa) non funziona correttamente. Come mai?
(Pierluigi Baglioni - Roma)

• Di solito il C/128 (ed i computer in generale) funzionano perfettamente sia con le periferiche collegate ed accese, sia spente.

Non dobbiamo dimenticare, però, che il collegamento tra elaboratore e periferica è garantito dal cosiddetto "bus" che, in pratica, è il cavo elettrico che lega tra loro computer e accessori.

Su ciascuno dei fili che costituiscono il bus è presente un segnale che, a seconda del suo valore, attiva particolari funzioni o procedure.

Un filo "spento" non è sempre "insignificante" (come si potrebbe ritenere a prima vista) ma può rappresentare, addirittura, un ben preciso comando per l'intero sistema.

In altri casi la mancanza di un segnale ben preciso, che invece è riscontrabile con la periferica accesa, può provocare malfunzionamenti di varia natura.

Si pensi che, più spesso di quanto non si creda, la mancanza di una adeguata "messa a terra" può provocare guai difficilmente individuabili.

Su qualsiasi manuale, del resto, è sempre consigliata l'accensione dell'apparecchio e la sua messa a terra. Si deve ritenere, insomma, che l'eccezione è rappresentata dal sistema che funziona correttamnte nonostante alcune apparecchiature siano spente, e non il viceversa!

CARICARE IL CP/M

□ Come devo agire per caricare il CP/M con il mio C/128?
(Massimo Martelli - Ostellato)

• Spegnere computer e drive. Accendere il drive ed inserire il dischetto CP/M. Accendere il C/128; attendere alcune decine di secondi per veder apparire il più obsoleto e preistorico sistema operativo del mondo.

RASTER SINISTRO

□ Grazie al Raster register è possibile visualizzare, contemporaneamente, una parte della pagina testo ed una parte di quella hi-res. E' possibile miscelare le due pagine nel verso destra - sinistra invece che alto - basso?
(Mirko Lalli - C.na Stazione)

• Tutte le tecniche legate alla manipolazione del Raster si basano sulla possibilità di selezionare, sul video, una "riga" alta un pixel (puntino elementare) e lunga 320. Ciò è dovuto al fatto che il pennello elettronico traccia una riga orizzontale prima di passare a quella successiva.

Siccome il video è lungo 40 caratteri, il numero di "colonne" elementari (ognuna corrispondente ad un pixel) che è possibile visualizzare è 320 (=8 x 40). Ciò significa che, con particolari tecniche, si riesce a creare una corrispondenza tra una riga elementare e 40 byte posti di seguito nella memoria. Se questi byte appartengono alla pagina grafica, alla mappa dei caratteri o ad una qualsiasi altra zona (Ram o Rom) poco importa per il pennello elettronico: questo terrà acceso, oppure spento, il pixel sulla riga video a seconda se è incaricato di segnalare, rispettivamente, un bit settato (posto a 1) oppure a zero.

La manipolazione del raster può spingersi fino a controllare una riga elementare alla volta, purchè INTERA. Non è possibile individuare la posizione del pennello elettronico all'interno della riga stessa; tale condizione, invece, dovrebbe esser necessaria per realizzare quanto richiesto.

C'E' NESSUNO IN CASA?

□ Come è possibile sapere se, su disco, è presente un certo file?
(Giuseppe Cannella - Palermo)

LA VOCE DEI LETTORI

AMIGA

Per ciò che riguarda Amiga, che ha sostituito il mio C/64, vi suggerisco di ripercorrere (in "chiave" Amiga) le tappe già seguite per il glorioso computer precedente: approfondire la conoscenza della macchina, periferiche, files, gestione della memoria, che cosa ha sostituito le Sys, come vengono usate le Peek e le Poke e così via. Un'analisi dettagliata del contenuto dei due dichi "ufficiali" (Workbenc e Extras), qualche buon listato Basic ed altre cose del genere farebbero la felicità di noi possessori di Amiga. Avete pensato a pubblicare un'intera rivista dedicata a questo straoridinario computer?

LA VOCE DEI LETTORI

PLUS-4

Volendo sapere come venire in posseso del manuale del programmatore per il Plus-4 (Programmer's Reference Guide), ho telefonato alla Commodore Italiana: non sapevano nemmeno che la Commodore avesse prodotto un computer con questo nome! Ho scoperto come comandare il bus dei dati della porta parallela "ma se la tecnica non è applicabile al C/64 non ci interessa" rispose una voce della vostra Redazione. Tutti deridono il Plus-4, ma non sanno che lo sfruttamento di tutte le sue risorse farebbe tremare i piedi del "nonno" 64. La conseguenza è che ho deciso di acquistare un computer di marca NON Commodore.
(Roberto N. - Bologna)

• Il problema si risolve tentando di "aprire" quel file in lettura e di esaminare la "risposta" fornita dalla solita subroutine che gestisce il canale di errore n. 15. Una soluzione può esser la seguente:

```
100 open 15, 8, 15
105 open 1, 8, 8, "nome"
110 input #15, a$, b$, c$, d$
120 print a$; b$; c$; d$
130 close 8: close 15
```

"Nome", ovviamente, deve essere il nome del file di cui si desidera scoprire l'esistenza. Il file puo' esser qualunque (seq, prg, usr, rel) e non è quindi necessario specificarne la natura.

Opportune istruzioni (if a$=...) dovranno, come intuitivo, dirottare il programma verso le decisioni da prendere a seconda del caso verificatosi.

Se si desidera individuare, oltre al nome, anche il tipo di file, è sufficiente modificare la riga 105...

```
105 open 1, 8, 8, "nome, s"
105 open 1, 8, 8, "nome, p"
105 open 1, 8, 8, "nome, r"
```

...oppure...

```
...u$  ".s"
105 open 1, 8, 8, "nome" + u$
```

...e, in base alla risposta fornita, stabilire quanto desiderato.

SFORZI ECCESSIVI

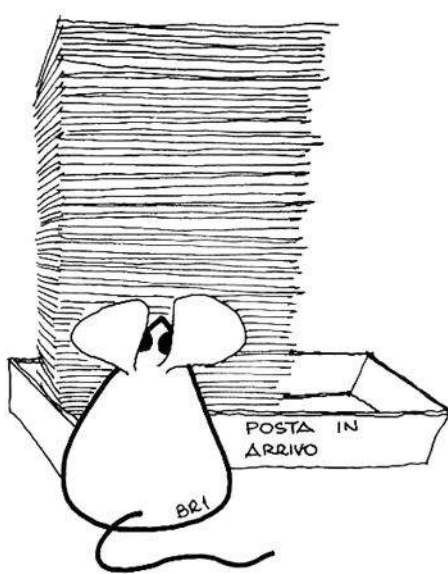
□ Un tecnico che ripara TV ha affermato che l'uscita video del C/64 mette sotto sforzo il televisore e che devo utilizzarlo per non più di un'ora per volta; inoltre, dopo l'uso, dovrei tenere spento il TV per almeno due ore. E' possibile una cosa del genere?
(Renato B. S.te Olona)

• A quel tecnico non affiderei nemmeno il compito di cambiare una lampadina bruciata. Sei sicuro che non volesse, per scherzo, prenderti in giro?

HARD COPY C/64

□ Mi piacerebbe veder pubblicata questa brevissima routine che, seppur lenta, risolve il problema dell'hard copy dello schermo in bassa risoluzione.
(G. Falco - Roma)

• Eccoti accontentato:

```
100 for i=1 to 999
105 print chr$(rnd(0)*64 + 64);:next
110 gosub 63000:end
62970 :
62980 rem hard copy low res
62990 :
63000 open 4, 4: x=1024
63010 for s=1 to 25: for t=1 to 40
63020 a=peek(x): x=x+1
63030 a=a+128* (a 〉 127)
63040 a=a-64*(a 〈 32 or a〉95)-32*(a 〉
63 and a 〈 96)
63050 print#4, chr$(a);
63060 next t: print#4: next s: close4
63070 return
```

La routine, ovviamente, è solo quella compresa tra le righe 63000 e 63060; le righe 100 - 110 sono solo un demo.

La tecnica adoperata considera la corrispondenza tra il codice schermo dei vari caratteri ed il corrispondente codice Ascii della stampante. La routine, scritta per il C/64-128, dovrebbe funzionare anche con il C/16 e Plus/4 modificando in X=3072 l'istruzione di riga 63000.

COLORANDO

□ I due micro programmi che allego si comportano in modo per me misterioso ed errato. Potete darmene una spiegazione?
(Livio Ariboli - Arezzo)

• "Se provate (continua il nostro lettore) a digitare il seguente listatino, ed a premere Run/Stop e Restore (per ripristinare le condizioni iniziali)...

```
5 Print Chr$(5) :rem colore bianco
6 Print Chr$(147): rem cancella schermo
7 Poke 1480, 1: rem cod. video "A"
8 Print "A"
```

...dopo il Run compaiono, in alto a sinistra ed al centro, due caratteri "A" entrambi bianchi. E fin qui tutto è normale.

Modificando il programma in...

```
5 Print Chr$ (147)
6 Print Chr$ (5)
7 Poke 1480, 1
8 Print "A"
```

...e cioè invertendo le sole prime due righe, dopo aver premuto nuovamente Run/Stop e Restore (per ripristinare le condizioni ini-

LA VOCE DEI LETTORI

VIVA IL PLUS-4

Vorrei puntualizzare alcune cose riguardo al Plus-4. Sul N. 49 avete affermato che non esiste alcun compilatore per il sopracitato computer. Ebbene, io dispongo da tempo di Austrospeed (lungo 124 bocchi), versione specifica per Plus-4 derivata da quella, omonima, per C/64. Sul N. 40 e 48 avete affermato che non esiste alcun sintetizzatore vocale per Plus-4. Ebbene dispongo da tempo di "Speech Plus" che ritengo di buona qualità.

Queste precisazioni mi sembrano necessarie per la semplice ragione che i possessori dell'ottimo, ma sfortunato, Plus-4 hanno diritto ad una corretta informazione sul software disponibile, almeno da parte vostra. La difficoltà di reperire programmi validi sono notevoli; se, poi, anche voi informate male (spero involontariamente!) allora proprio non ci siamo...

Spero che questa mia lettera ravvivi l'entusiasmo e l'interesse di molti lettori verso il loro computer, per il quale dispongo di molto altro software utile (Pascal, gestione archivi, fatturazione, eccetera).

Mi piacerebbe entrare in contatto con utenti (seri) del Plus-4, anche se residenti all'estero.

Alessandro Salviato
Via Delle Rose, 9
31033 Castelfranco Veneto

ziali) compare, dopo il Run, una "A" bianca in alto a sinistra, ma una "A" celeste al centro. Non si dovrebbe, invece, avere lo stesso risultato di prima?"

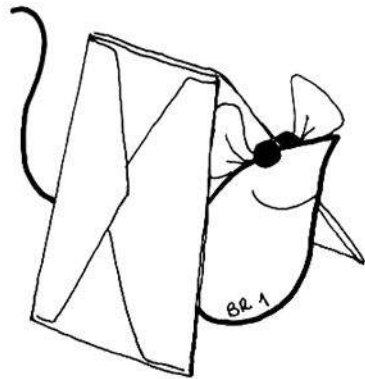
No, caro Livio, ed il motivo è presto detto: quando lo schermo viene cancellato con i tasti Shift e Clr/Home, oppure con Chr$(147), ogni cella del video verrà colorata con il colore che il cursore aveva PRIMA di impartire il carattere di cancellazione schermo.

Nel primo caso, dunque, al momento della cancellazione è già attivo il colore bianco (riga 5) e la Poke di schermo colora in tal modo la lettera "A".

Nel secondo caso, invece, quando lo schermo viene cancellato, il cursore è di colore celeste, colore standard che si ottiene premendo Run/Stop e Restore.

Prova a far girare il secondo listatino colorando (tasti Control 1) di nero il cursore: la "A" centrale resterà nera; e così via.

STAMPANTI RS-232

□ Possiedo un C/64 ed una stampante dotata di standard Rs-232. Mentre con listati da me realizzati stampo senza problemi, con programmi professionali ciò non mi è possibile. Come posso fare?
(Maurizio Fuschetto - Napoli)

• La stampante è "vista", dal C/64, come periferica n. 4 mentre gli accessori collegati alla porta seriale sono contraddistinti, altrettanto automaticamente, dal n. 2. Purtroppo il C/64 non dispone della (comodissima) opzione della redirezione della stampante che consente di assegnare un qualsiasi numero "logico" ad una qualsiasi periferica. Tale opportunità è invece offerta dai sistemi "superiori", come l'Ms-Dos.

Non ti resta che usare programmi professionali (come Easy Script) che, prevedendo la connessione con apparecchiature diverse da quelle standard, permettono di effettuare collegamenti senza problemi di sorta.

PLOTTER E HI-RES

□ Posseggo il mini-plotter 1520 e mi piacerebbe entrare in possesso di una routine di hard copy della pagina grafica.
(Arrigo Frigeni - Milano)

• Le comuni routine di hard copy sfruttano la notevole affinità esistente tra la gestione del video hi-res e la caratteristica di funzionamento delle stampanti ad aghi.

Infatti ad ogni pixel "acceso" del video si fa corrispondere la "spinta" di un ago (detto, più propriamente, "dot") che riproduce il puntino sul foglio di carta, riga dopo riga.

Un plotter, invece, funziona in modo del tutto diverso ed è più idoneo a funzionare con sistemi di grafica vettorizzata.

Ciò significa che i programmi grafici destinati ad attivare un plotter, benchè visualizzino su schermo una immagine del tutto identica a quella ottenibile con altre tecniche di riproduzione su video, memorizza, in opportune matrici, gli estremi dei segmenti tracciati o delle figure che è possibile parametrizzare (rettangoli, cerchi, parabole, eccetera).

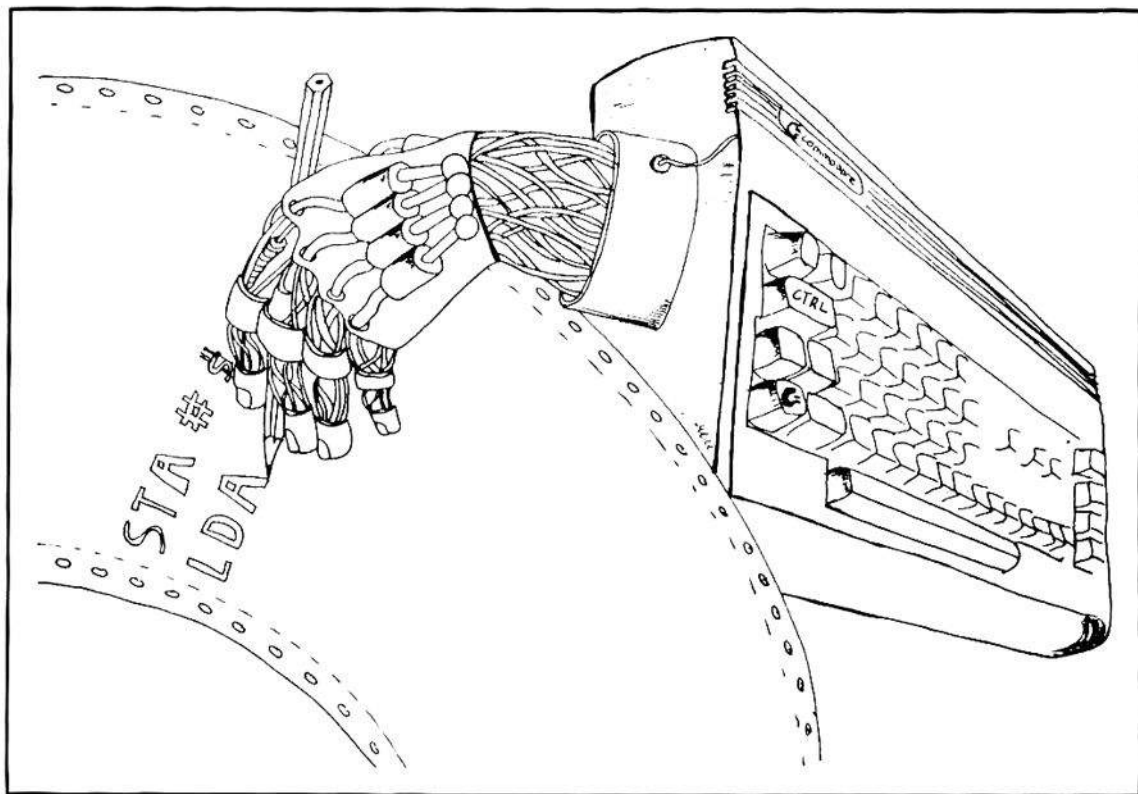
Al momento della stampa il computer non invia la pagina grafica bit per bit (come in una comune hard copy per stampante ad aghi) ma provvede ad inviare, appunto, gli elementi della matrice che, opportunamente interpretati dal "cervello" del plotter, generano il disegno con una precisione decisamente superiore a quella ottenibile da una comune stampante ad aghi (per non parlare della possibilità di cambiar colore).

In linea di massima, tuttavia, non è impossibile usare, con un plotter, la stessa tecnica di riproduzione di una stampante ad aghi; vi sono, però, due inconvenienti. Il primo è la lentezza (occorrerebbe una ventina di minuti per una hard copy!) ed il secondo è il superlavoro cui verrebbe costretta la periferica, specie se il disegno è formato da molti chiaro-scuri.

In ogni caso non è assolutamente sufficiente cambiare il numero di periferica da 4 (standard 803) in 6 (standard 1520) dal momento che le tecniche di gestione dei due accessori sono profondamente diversi.

COMPETENZE

Ritengo che il più grosso problema in Italia, dopo la pirateria, è l'incompetenza dei rivenditori di computer. In molti dei negozi di informatica di Milano, specializzati nella vendita di prodotti Commodore, non hanno saputo darmi spiegazioni esaurienti su alcuni accessori per il mio C/64, nè sapevano dell'esistenza del "Macro Assembler Commodore". A chi affidano la vendita di computer e, soprattutto, perchè si afferma che il personale viene "severamente selezionato ed addestrato" se, poi, non sanno nemmeno ciò che vendono?

(Diego K. Melegnano)

PRONTO, CHI STAMPA?

Non sempre i termini riportati sui manuali delle stampanti sono chiari; esaminiamone alcuni

di **di Alessandro de Simone**

Chi si avvicina ad un computer per la prima volta viene preso, di solito, da un certo sconforto: termini tecnici sconosciuti, connessioni multiple, precauzioni nel digitare dati, elaboratori che si bloccano senza apparente motivo....

Figurarsi che cosa accade quando si decide di acquistare una periferica e, in particolare, una stampante: di questi accessori, infatti, ve ne sono di ogni tipo e, come intuitivo, di ogni prezzo. Vi aiutiamo, con le brevi note che seguono, a districarvi tra i numerosi termini tecnici e, soprattutto, a capire quale è il tipo di stampante adatto alle vostre esigenze.

• AMPIEZZA DI PAGINA

E' il valore (in pollici o centimetri) che rappresenta le dimensioni massime, orizzontali, del foglio di carta che può essere inserito in una stampante. In quelle più economiche è possibile inserire il formato A4 (21 x 29.6) con il lato minore parallelo alla base; le stampanti in grado di ospitare un formato doppio (A3) sono utili negli studi professionali in cui è necessario inserire, tra l'altro, fogli protocollo "aperti" o tabulati di dimensioni ragguardevoli (soprattutto elaborazioni di spreadsheet).

E' ovvio che è possibile inserire anche fogli, e moduli continui, di dimensioni minori rispetto alla misura massima consentita; in particolare si ricorda la disponibilità, in commercio, di etichette autoadesive in "strisce" (ideali per stampare indirizzi), di moduli continui contenenti buste, di carta legale rigata, di fogli dotati di carta carbone facilmente estraibile e così via. Di solito anche le stampanti supereconomiche consentono l'utilizzo di tali moduli, fatte salve le dimensioni massime consentite.

• BIM (Bit Image Mode)

E' la modalità grafica attivabile dalla quasi totalità delle stampanti moderne. Siccome

la testina dispone di una certa quantità di aghi (posti l'uno sotto l'altro, in una fila verticale), l'invio preventivo di un particolare comando (di solito ESC K) consente di programmare l'impatto di ciascun ago in una delle 640 file verticali possibili sul rigo stampabile (80 caratteri x matrice di 8 = 640). Il numero di punti per pollice, tuttavia, può esser variato in molti modi, grazie al solito invio del carattere di comando riportato sui libretti di istruzione. Le stampanti a 16 e 24 aghi, in alcuni casi, sembrano offrire la stessa risoluzione grafica di quelle a 9 aghi; ciò dipende, però, dal software che viene utilizzato: alcuni programmi professisonali, infatti, attivano le stampanti a 24 aghi in modo 9 aghi e la BIM sembra eguale nei due casi.

• BUFFER

Il computer, di solito, è molto più veloce della stampante dal momento che il primo opera a velocità elettronica, la seconda, invece, deve fare i conti con parti elettromeccaniche in movimento, nettamente più lente. Ciò significa che se devono essere inviati, ad esempio, 400 caratteri (equivalenti a 5 righe di 80 caratteri ciascuna) la stampante, durante la riproduzione su carta dei primi 80 caratteri, è costretta ad inviare, al computer, un segnale (busy = occupato) che ha il compito di interdire l'invio di ulteriori dati. L'elaboratore, pertanto, rimane in attesa di inviare il rigo successivo (cioè il secondo) nonostante sia già in grado di inviarlo. Quando la testina ha terminato di stampare il primo rigo, e si è posizionata all'inizio di quello successivo, invia al computer il segnale di "via libera" ed il secondo rigo viene quindi inviato, accettato e riprodotto su carta; così si procede per i rimanenti. Ciò accade, come intuitivo, se la stampante ha un buffer (= memoria temporanea di transito) di 80 Ram byte, nella quale allocare i dati provenienti dal computer. E' facile immaginare che se il documento da stampare è lungo diverse migliaia di byte il computer è costretto a sospendere ogni altra attività finché non "scarica" l'intero documento, operazione ostacolata dalla lentezza della periferica. Alcune stampanti dispongono di buffer decisamente più ampi (anche parecchie migliaia di byte), in questi casi l'invio, e la successiva allocazione dei dati (estremamente veloce), libera il computer dalle inutili attese, e l'elaboratore rimane quindi disponibile per effettuare altre operazioni.

• CARICAMENTO AUTOMATICO

Spesso si ha l'esigenza di stampare fogli personalizzati dall'utente (lettere intestate, moduli prestampati, eccetera), questi, nella generalità dei casi, sono disponibili non in modulo continuo, ma in pacchi di fogli singoli. La scomodità di inserire, manualmente, un foglio alla volta può essere eliminata usando un apposito accessorio (di solito fornito di serie) che, ospitando diverse decine di fogli singoli, libera l'utente dal fastidio di introdurli uno alla volta. L'inserimento, infatti, risulta totalmente automatico.

• CODICI DI CONTROLLO

Alcuni caratteri "speciali", che non corrispondono a simboli riproducibili graficamente, sono interpretati, dalla stampante, come comandi da eseguire. Il più noto è il comando ESC [Chr$(27)] che, seguito da uno o più caratteri alfanumerici, rappresenta un comando ben preciso. Tra questi ricordiamo le varie impostazioni dei modi di stampa, font di caratteri, interlinee, spaziature, allargamenti, sottolineature e così via. Le opportunità offerte dai comandi speciali sono sfruttate dai più importanti pacchetti di software professionale, in special modo word processor e data base. Se una stampante non dispone di un certo codice, e si tenta di inviarlo egualmente, l'effetto può essere nullo oppure la riproduzione di caratteri indesiderati.

• CPI

E' l'acronimo di Character Per Inch (= caratteri per pollice); indica il numero di caratteri che è possibile affiancare, orizzontalmente, in un pollice (= 2.54 cm.). Più piccoli sono i caratteri, maggiore sarà il loro numero. Ad ogni tipo di carattere corrisponde un ben preciso valore di CPI. Esempio: Pica (10); Elite (12); Condensed (17.1). Il CPI può essere impostato manualmente (settando alcuni interruttori prima di accendere la stampante) oppure via software, grazie all'invio dei "soliti" caratteri di controllo. Spesso alcuni modelli dispongono di opportuni pulsanti, posti sul frontalino, che permettono la modifica dei CPI anche durante la stampa di un documento.

• DIP SWITCH

Molte stampanti offrono la possibilità di impostare, via hardware, alcuni modi particolari di stampa. Ciò si realizza posizionando su ON, oppure su OFF, alcuni dip switch (micro-interruttori). Tale "regolazione" va eseguita prima di accendere la stampante altrimenti, se effettuata in seguito, può non essere "riconosciuta". Di solito tutte le impostazioni effettuabili via hardware possono essere realizzate anche via software, mediante l'invio di determinati codici di comando. La comodità dei dip switch, quindi, risiede nel non esser costretti ad impostare via software (tutte le volte che si accende la stampante stessa) la particolare configurazione che si è deciso di usare con maggiore frequenza.

• DIREZIONE

Le stampanti moderne offrono la cosiddetta bidirezionalità. Questa consiste nello stampare il primo rigo da sinistra a destra, quando la testina è arrivata in questa posizione (estrema destra) il foglio viene fatto avanzare ed il rigo successivo viene stampato, da destra a sinistra, a partire (ovviamente!) dall'ultimo carattere. In questo modo vengono eliminati i tempi morti che le vecchie stampanti, invece, richiedevano affinché la testina di stampa "tornasse" all'estrema sinistra. Le moderne stampanti, inoltre, offrono il percorso bidirezionale "ottimizzato". Con questo termine si intende, in pratica, la possibilità di eliminare altri tempi morti, impedendo il percorso di porzioni di righe in cui dovrebbero essere stampati spazi... vuoti. In modo grafico quasi sempre, viene attivata la monodirezionalità anche in stampanti che offrono la bidirezionalità ottimizzata.

• IC CARD

Alcune stampanti offrono la possibilità di scrivere con altri font (tipi) di caratteri. Di solito vengono offerte particolari schede elettroniche, grandi più o meno quanto una carta di credito, da inserire nell'apposito connettore. In tali IC Card possono trovare posto altri font, oppure un buffer di memoria Ram, spesso di notevole capacità, in grado di elevare, in pratica, la velocità di trasferimento dei dati; in questo caso la scheda prende più propriamente il nome di RAM CARD. (cfr. Buffer)

• INTERFACCIA

Un computer si collega alla stampante mediante un cavo elettrico particolare. Ogni periferica (ed anche il computer) dispone, al suo interno, della circuiteria elettronica idonea alla rice-trasmissione dei dati. Tale circuiteria prende il nome di "interfaccia" e ve ne sono di diversi tipi. Il più usato (nel caso delle stampanti) è il sistema Centronics che, inviando un intero byte (formato da 8 bit) alla volta, risulta piuttosto veloce ma richiede un cavo contenente numerosi fili elettrici. Un'altra interfaccia,

ma di tipo seriale (che, al limite, può essere costituita da soli due cavi), è la RS-232 che consente, però, l'invio di un bit per volta e risulta quindi più lenta della Centronics. Tale interfaccia è insostituibile nel caso di trasmissione di dati via modem, ove non sia disponibile un bus (cavo elettrico) formato da più di due fili, come è il caso del telefono. Un'altra interfaccia è la parallela IEEE-488, usata quasi esclusivamente per collegare un computer con strumenti di misura. I piccoli Commodore (C/64, C/128, C/16) richiedono un'interfaccia specifica, creata apposta dalla Commodore. Tale interfaccia si trova sulle stampanti cosiddette "803 compatibili" che, pertanto, devono soddisfare pienamente le caratteristiche del protocollo di comunicazione Commodore.

• INTERLINEA

E' la distanza che intercorre tra due righe di stampa successive. Può essere impostata mediante gli appositi comandi dei programmi professionali, oppure grazie all'invio (h/w oppure s/w) di determinati codici. L'interlinea e la forma stessa dei caratteri sono causa della diversità dell'aspetto di un documento tra come appare sul video e come viene, invece, riprodotto su carta.

• LETTURA FACILITATA

I fogli di carta sono posti in vendita, oltre che totalmente bianchi, anche in un formato particolare, costituito da righe orizzontali colorate (di solito in celeste) che si alternano a righe bianche. Questo formato favorisce l'agevole lettura soprattutto di lunghi (e... larghi) tabulati in cui centinaia di dati possono essere ospitati su un solo rigo. In tal modo, infatti, la diversa colorazione di due righe successive "guida" lo sguardo del lettore che riesce meglio ad interpretare i dati stessi, eliminando il pericolo di esaminare, per errore, i dati presenti sulla riga precedente o su quella successiva. Nel caso di normali testi, o di riproduzione di grafici, la carta con lettura facilitata può costituire, al contrario, una difficoltà di lettura.

• LUNGHEZZA DI PAGINA

I pacchi di carta in commercio differiscono tra loro, oltre che per la larghezza, anche per la lunghezza dei fogli. Per fare in modo che la stampante "salti" automaticamente la parte tratteggiata (che separa un foglio dal successivo) è indispensabile impartire (via h/w oppure s/w) l'apposito comando relativo alla lunghezza della pagina. Inutile dire che moltissime stampanti in commercio consentono l'impostazione, via software, anche della lunghezza della pagina.

• NASTRO INCHIOSTRATO

E' inutile spiegare che cosa sia. Importante, invece, è documentarsi, prima dell'acquisto della stampante, sulla reperibilità (e prezzo) dei nastri di ricambio!

• NLQ

E' l'acronimo di Near Letter Quality(= qualità simile a quella di una macchina da scrivere). Si realizza stampando un rigo in due fasi successive: la prima volta i caratteri vengono stampati normalmente; la seconda "passata" viene quindi ripetuta facendo avanzare di una frazione di millimetro il foglio di carta. In tal modo la matrice del carattere, formata dall'impatto degli aghi, risulta meno visibile ed il testo appare più netto. La velocità di stampa in modo NLQ, ovviamente, è minore di quella in modo Draft (normale). Le stampanti a 16 o 24 aghi, grazie alla disponibilità di un maggior numero di aghi, consentono di ottenere il modo NLQ in una sola passata e risultano, di conseguenza, molto più veloci dei modelli a 9 aghi.

• ON / OFF LINE

Tutte le stampanti dispongono di questo pulsante che se premuto, impedisce l'invio di altri caratteri da parte del computer che rimane, pertanto, in attesa che il pulsante venga premuto nuovamente. E' molto utile quando sia necessario cambiare o posizionare un foglio, sostituire un rullo di carta esaurito e così via. Il modo Off Line viene automaticamente impostato in caso di assenza di carta o nel caso in cui questa si esaurisca. A volte il non corretto inserimento di fogli di piccole dimensioni impedisce al sensore della carta di "accorgersi" della presenza del foglio e la stampante permane inspiegabilmente inattiva.

• RESA GRAFICA

E' la possibilità, offerta da una stampante, di miscelare tra loro modalità di scrittura (soprattutto Draft e NLQ), spaziature orizzontali (Pica, Elite, Condensed e così via) e versioni (Grassetto, Doppia larghezza, Sottolineatura, eccetera).

• SET DI CARATTERI

E' la capacità di riprodurre caratteri Ascii, grafici e speciali.

• SISTEMA DI AVANZAMENTO

Il tipo di carta più usato dagli hobbysti è certamente quello con i bordi traforati posti ai lati del foglio stesso, il cosiddetto "modulo continuo". Per far avanzare un modulo continuo è necessaria la presenza di un trattore, vale a dire la coppia di ruote dentellate che, agganciando i fori laterali, fanno avanzare la carta. L'altro sistema di avanzamento è quello a frizione che, serrando il foglio tra due rulli di gomma, costringe la carta ad avanzare grazie alla loro rotazione. Le stampanti che dispongono del solo avanzamento a frizione possono ospitare anche fogli forati; il viceversa, invece, non è possibile. In commercio, comunque, non esistono più stampanti che offrono, come modo di avanzamento, il solo trattore.

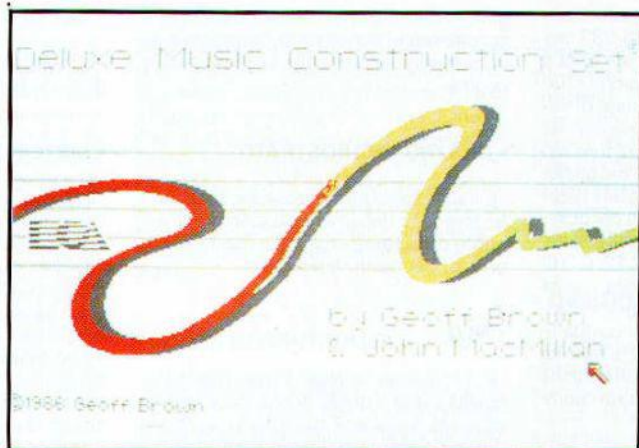
• TECNICA DI STAMPA

Può essere ad impatto, ink-jet e laser. Nel primo caso i vari caratteri vengono riprodotti grazie all'impatto generato da una, due oppure tre file di aghi contenuti nella testina di stampa. Nel secondo caso al posto degli aghi sono presenti micro-ugelli dai quali fuoriescono getti di inchiostro che si depositano sulla carta. Le laser printer, infine, usano la medesima tecnica delle fotocopiatrici. La rumorosità diminuisce passando dal primo all'ultimo metodo di stampa. Il prezzo di vendita, invece, segue l'andamento diametralmente opposto.

• TEST

Tutte le stampanti dispongono di un metodo che consente un (pur se sommario) esame delle funzionalità della periferica stessa. Si tratta dell'autotest, che viene attivato accendendo la stampante mentre il pulsante Line Feed è tenuto abbassato.

• VELOCITA' DI TABULAZIONE

Molti fabbricanti, per aumentare la velocità di stampa, adottano un accorgimento particolare per cui la testina percorre eventuali spazi vuoti ad una velocità maggiore del normale. La riproduzione del documento risulta quindi più veloce, soprattutto nella stampa di tabulati che posseggono numerosi spazi vuoti al loro interno.

DELUXE MUSIC CONSTRUCTION SET

Uno straordinario "pacchetto" musicale che può trasformarvi in abilissimi compositori

di **Luigi Callegari**

Deluxe Music Construction Set è stato uno dei primi programmi per Commodore Amiga ed ha subito numerosi miglioramenti e modifiche da quando la Electronic Arts lo introdusse nel mercato americano nel 1986.

Si tratta, essenzialmente, di un programma per la composizione musicale su pentagramma, interfacciabile con strumenti MIDI previo acquisto dell'interfaccia necessaria (vedi inserzionisti di CCC). La composizione può poi essere trasferita su carta tramite una stampante grafica, con tutte le convenzioni (e gli orpelli grafici...) care ai musicisti.

IL PACKAGE

Al prezzo di circa novantamila lire viene fornito un dischetto standard Amiga ben protetto contro i pirati ed un manuale d'uso in inglese (presto tradotto in italiano dalla CTO) di circa 70 pagine con varie illustrazioni.

La configurazione minima richiesta è Amiga e basta, ma è bene avere un secondo drive se si vogliono evitare troppi scambi quando si lavora, nonchè una espansione se si vogliono creare spartiti molto lunghi e con strumenti sofisticati; una stampante grafica non guasta se si desidera l'output degli eleganti spartiti generati.

L'uso è completamente controllato da mouse, menu a discesa e numerose finestre.

INIZIAMO

Per incominciare si può provare a caricare un file tra i dimostrativi presenti sul disco nella directory SCORES. Si usa il menu FILE e l'opzione OPEN SCORE, ricordando che NEW SCORE crea, invece, un nuovo spartito. Per salvare lo spartito con lo stesso nome si usa SAVE mentre SAVE AS lo registra sul disco con il nome specificato.

Usando SAVE AS SMUS FILE lo spartito può venire registrato in formato standard IFF (Interchange File Format) e quindi essere scambiato con altri programmi musicali. Con REVERT si riottiene il file di spartito come era stato memorizzato su disco, prima che incominciassimo a redigerlo.

Una volta caricata la musica, questa può essere redatta col topo e con la tastiera raffigurata sul video. Per effettuare alterazioni bisogna prima selezionare un blocco di note di un rigo, clickando sulla prima ed ultima nota. Con l'opzione SELECT ALL si opera contemporaneamente sull'intero spartito. Il blocco selezionato può essere tagliato (CUT), cancellato (CLEAR) oppure ricopiato nel buffer temporaneo (clipboard) con COPY e quindi inserito, come con un word processor, tramite un PASTE nel punto in cui si è collocato il cursore del mouse. Con l'opzione UNDO, si annullano gli esiti dell'ultimo comando eseguito.

Le note (e tutti i simboli di pausa, diesis, eccetera) si inseriscono clickandoli nella finestra laterale dello spartito ed inserendoli quindi col mouse.

IL MENU WINDOW

Con il menu "window" si controllano le varie finestre di DMCS. La finestra chiamata nei menu SCORE è quella contenente lo spartito. PIANO KEYBOARD contiene una tastiera di sei ottave suonabile col mouse e NOTE PALETTE contiene i simboli grafici inseribili nello spartito (pause, note, alterazioni e gruppi irregolari di terzine e cinquine, segni dinamici).

Tutte le finestre sono manovrabili, come al solito in Amiga, tramite il mouse.

Un piccolo difetto del programma è che lavora solo in modo NTSC (e non in PAL); non è quindi possibile usare le 56 linee supplementari dei nostri Amiga europei. Le finestre sono presenti per default, mentre SCORE SETUP può essere richiamata dal menu per regolare molti parametri generali dello spartito. La porzione superiore

L'HARDWARE SONORO DI AMIGA

Amiga dispone di quattro canali sonori hardware separati, ciascuno programmabile indipendentemente dagli altri. Tali canali sono collegati a due connettori e possono essere combinati per generare suoni stereofonici indipendenti o contemporanei su due casse acustiche (collegate ad un impianto HI-FI od ad un monitor adeguato).

I canali possono esser collegati tra loro in modo che uno moduli il suono generato dall'altro, in modo da ottenere effetti "creativi". Ciascun canale audio comprende un convertitore digitale-analogico ad otto bit controllato direttamente dal sistema DMA del computer (accesso diretto in memoria, senza ausilio del processore 68000). Tale sistema, operante a 14 Mhz (contro il singolo mega del C/64 che dispone, poveretto, solo dell'interrupt...), consente al chip sonoro di leggere i dati di due forme d'onda complete durante ogni scansione di quadro televisivo. Ciò significa che la "voce" di due canali, ad esempio uno strumento oppure un suono digitalizzato, possono essere cambiate fino a cento volte al secondo. Nel caso si usino voci semplici (strumenti tipo pianoforte od organo) il DMA può memorizzare e ripetere una forma d'onda infinitamente, senza alcun ausilio del processore.

permette di modificare il numero di misure per riga, il tempo di esecuzione (battute / minuto), la larghezza in pixel del rigo, il volume sonoro di riproduzione, il modo di scorrimento dello spartito durante l'esecuzione, il numero di tracce per rigo, il flag di controllo della visualizzazione del numero di stile di esecuzione.

La porzione inferiore di SCORE SETUP controlla invece con varie opzioni i singoli righi, citati per numero sequenziale (sistema poco comodo ma necessario), assegnando una delle quattro possibili chiavi musicali e permettendo l'aggiunta di un rigo supplementare (detto staff) e l'attivazione, o meno, durante il playback. Inoltre si può indicare se si desidera visualizzare i nomi degli strumenti usati (cambiabili durante l'esecuzione, nei vari punti dello spartito) e la chiave musicale con relative tonalità. Dal momento che è possibile inserire testi nello spartito, da qui si può regolare anche lo spazio vuoto sotto e sopra il pentagramma.

Poichè sono ammessi sino ad otto righi, ciascuno dei quali con una o due tracce, è possibile avere sino a sedici partiture. Comuque, anche se Amiga internamente possiede solo quattro canali sonori, il software (e l'hardware supplementare a parte) consentono di collegare sino a sedici canali MIDI.

IL MENU PLAY

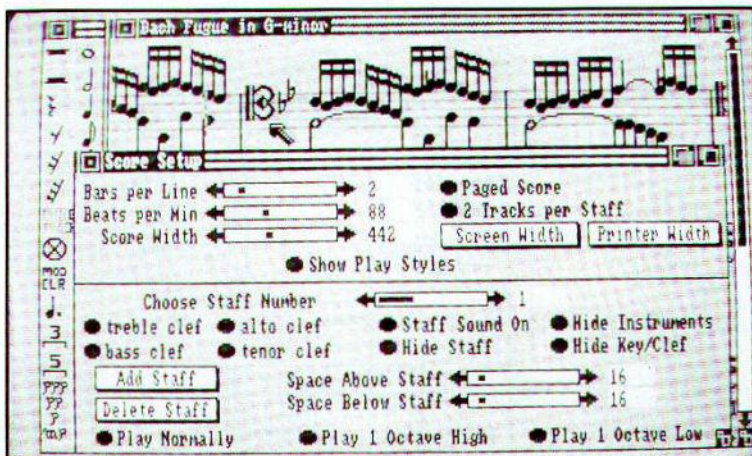
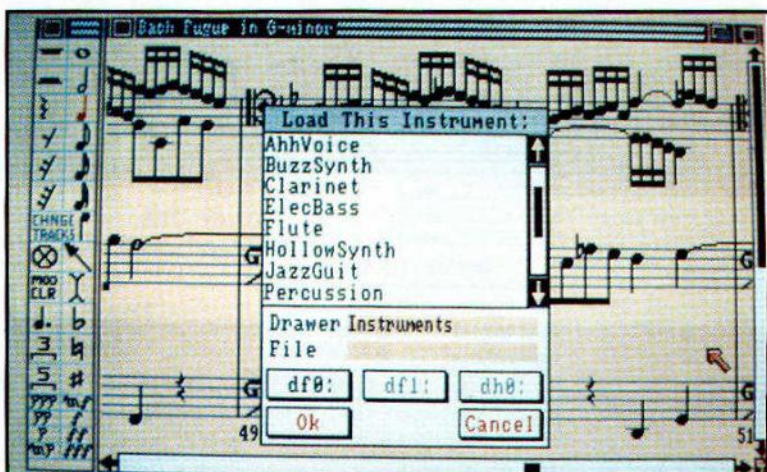
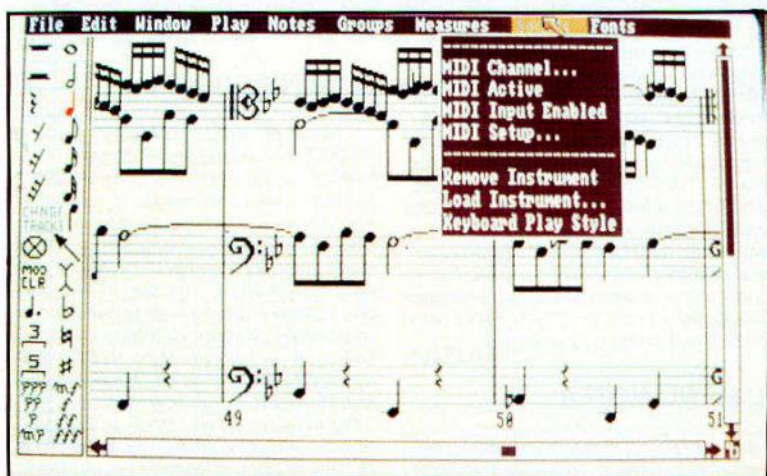
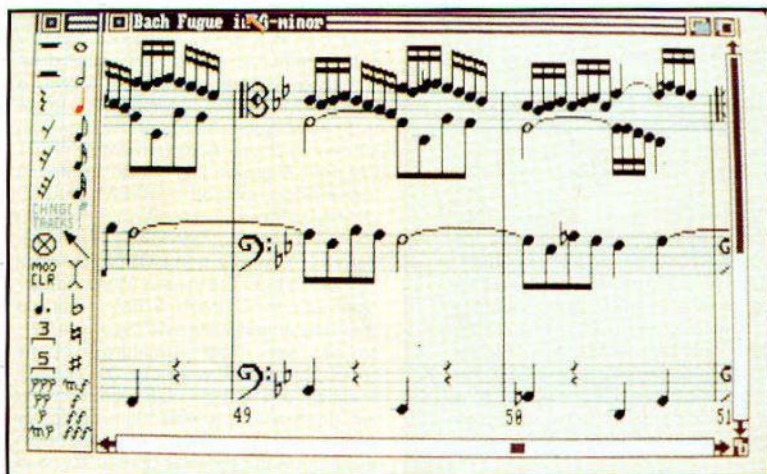
Il menu Play controlla l'esecuzione materiale del pezzo. PLAY SONG esegue brutalmente tutto lo spartito attuale mentre PLAY SECTION suona soltanto la porzione specificata col mouse e le opzioni BEGIN SECTION e END SECTION. La riesecuzione musicale può essere interrotta in ogni istante con l'opzione STOP PLAY o con la barra spaziatrice della tastiera, mentre per riprenderla si usa RESUME PLAY. L'opzione FLASH NOTES in riproduzione fa lampeggiare le note mentre vengono suonate, mentre TURN PAGE fa avanzare sullo schermo lo spartito mentre viene eseguito (cosa effettuata automaticamente se è attivo FLASH NOTES). PLAYER PIANO mostra "scenosamente" durante l'esecuzione i tasti corrispondenti alla nota riprodotta sulla tastiera a video. REPEAT PLAY ripete senza fine il brano o lo spartito.

IL MENU NOTES

Il menu NOTES presenta ben dodici opzioni che, guarda guarda, controllano le note selezionate. Le gemelle UP HALF STEP e DOWN HALF STEP consentono, rispettivamente, di alzare o abbassare di un semitono le note, mentre UP LEVEL e DOWN LEVEL alzano ed abbassano le note di una posizione (uno spazio od una riga) sul pentagramma. Analogamente, INVERT CHORD UP alza di una ottava la nota più bassa di un accordo e INVERT CHORD DOWN abbassa di una ottava la nota più alta di un accordo. Con HALF TIME si dimezza la durata e con DOUBLE TIME si raddoppia la durata di note. Molto elegantemente, con FLIP NOTE STEM si inverte la direzione attuale delle aste delle note (verso l'alto / verso il basso), mentre SET PLAY STILE apre una finestra che consente di scegliere una lista di stili di esecuzione da assegnare alle note selezionate, come "legato", "vibrato", "lento" e veloce. Solo questa opzione darebbe un bel "tocco in più" di professionalità a DMCS, introvabile in moltissimi altri programmi del genere.

IL MENU GROUPS

I comandi raggruppati sotto il menu GROUPS riguardano gruppi di note e la differenza rispetto a NOTES consiste anche nel fatto che qui si eseguono operazioni con simbologia adeguata sullo spartito. Con TIE NOTES si legano tra loro le note di identica altezza da eseguire come una sola nota. BEAM NOTES traccia un'asta nera che collega il blocco di note selezionato, se possibile; lo scopo di alcune opzioni di DMCS, si sarà capito, sono strettamente connesse alle potenzialità di generatore di spartiti, eleganti e graficamente efficaci.

Per assegnare il glissato al gruppo selezionato si usa SLUR NOTES UP o SLUR NOTES DOWN, ottenendo un passaggio

vellutato tra una nota e l'altra. Invece, le opzioni già tradotte (!) CRESCENDO e DECRESCENDO alterano gradualmente il volume di esecuzione di un gruppo di note (in crescendo o decrescendo, appunto) e sono anche usabili in congiunzione con i simboli di espressione (da pianissimo a fortissimo) per maggiore flessibilità. Per fare eseguire una ottava più in alto o più in basso un gruppo di note senza spostarle materialmente, si usano OCTAVE RAISE e OCTAVE LOWER.

IL MENU MEASURES

Tramite il menu MEASURES si agisce su intere misure o righi. Con SET TIME SIGNATURE, SET CLEF SIGNATURE e SET KEY SIGNATURE si alterano, rispettivamente, il tempo, la chiave e le alterazioni della chiave. Con SET INSTRUMENT si assegna lo strumento correntemente attivato nel menu SOUND nel punto corrente del rigo e SET TEMPO compie l'identica operazione usando invece dello strumento il tempo. Le due opzioni antagoniste delle precedenti sono ERASE INSTS e ERASE TEMPO, che annullano gli assegnamenti di strumenti e di tempo da un blocco di misure.

Vi sono anche numerose opzioni di manipolazione specifica diretta delle misure: INSERT MEASURE, SPLIT MEASURE, JOIN MEASURE, DELETE MEASURE, REALIGN MEASURE rispettivamente inseriscono, interrompono, fondono, sopprimono e riallineano misure.

Un altro gruppo di opzioni gestisce le ripetizioni e le doppie fini di partitura: BEGIN REPEAT, END REPEAT, 1st ENDING, 2st ENDING, DOUBLE BAR gestiscono le ripetizioni e le doppie finali delle partiture.

IL MENU SOUNDS

Il menu SOUNDS permette di gestire i timbri e gli strumenti. Sul dischetto sono presenti, in opportuna directory, una dozzina di strumenti. Nel menu è riportato l'elenco degli strumenti in memoria e quello usato attualmente è scritto in rosso. Usando SET INSTRUMENT si può specificare, a suon di topo, quale strumento usare sul rigo, che può poi essere rimosso usando REMOVE INSTRUMENT oppure caricato con LOAD INSTRUMENT.

Chi possiede interfaccia e strumenti Midi può usarli come se fossero strumenti-file

SOFTWARE SONORO

DMCS è solo uno dei tanti titoli presenti sul mercato (Sonix, Musicscape, Sound Oasis, Soundscape, Instant Music, Adrum, DxII, Datel Sound Studio, Synthia, Hot Licks...) per Amiga. E' un programma professionale e "classico", che però non sfrutta certamente appieno le caratteristiche sonore della macchina.

Si pensi che esiste un programma (oltre 300K in Assembly) specializzato a suonare la batteria elettronica (Adrum), con una sofisticazione incredibile. Altri programmi consentono di creare file sonori da usare in propri programmi e creati digitalizzando (tramite hardware apposito di basso costo) ciò che arriva dal nostro impianto HI-FI. Tutto grazie ad un hardware che conta forse solo un rivale (leggi Apple, ma posizionato al di sopra di qualche milione...) nel campo dei PC e, comunque, ancor più sorprendente tenendo conto anche delle altre caratteristiche extra della macchina, come la grafica ed il sistema operativo realmente multitasking.

Pensando al caro vecchio C/64 ci rendiamo conto di quali progressi siano stati fatti e guardando l'austero MS/DOS pensiamo con tenerezza al suo cicalino interno a voce singola.

letti da dischetto con relativi parametri. Si usa MIDI CHANNEL per aprire uno dei 16 canali disponibili, poi per abilitarlo effettivamente si usa MIDI ACTIVE e MIDI INPUT ENABLED mentre MIDI SETUP permette di regolare i parametri di interfaccia. L'opzione KEYBOARD PLAY STILE ricalca da vicino la già vista SET PLAY STYLE ma agisce ovviamente sull'interfaccia tastiera invece che sullo spartito.

Si noti che DMCS tratta sino a 16 canali MIDI, ciascuno dotato di un massimo di 128 regolazioni (preset).

NOTE VARIE

Un evidente difetto di un programma così professionale consiste nella mancanza dei sessantaquattresimi. Inoltre, se si richiede la visualizzazione dello spartito con lampeggiamento delle note e segnalazione sulla tastiera durante la riproduzione, si avvertono perdite di temporizzazione dovute a conflitti tra processori, specie se si dispone dei soli 512K standard (basta comunque disattivare tali opzioni scenografiche ma inutili).

La grafica del programma è personalizzabile, entro certi limiti, in quanto il programma legge file standard IFF (quelli generati da Deluxe Paint, Photon Paint eccetera), quindi chi ha il genio pittorico, oltre a quello musicale, può caricare il file contenente tali grafici, modificarli a piacere (lasciandoli ovviamente in ordine, sennò DMCS fa confusione) e poi risalvare il file su disco.

Quando si usa MIDI, bisogna evitare che due righi tentino di usare lo stesso canale sonoro.

A CHE SERVE?

Programmi come DMCS sono diventati strumenti di lavoro indispensabili per più di un compositore professionista, specie al giorno d'oggi, visto che la musica è sempre più elettronica, sequenziale, ripetitiva, sofisticata e, in una parola, computerizzata. Un programma del genere avrebbe entusiasmato artisti come Paul McCartney e John Lennon (pace all'anima sua), che non conoscevano la notazione musicale pur scrivendo melodie immortali firmate Beatles, in quanto consentono di comporre un testo con la tastiera (o meglio, con uno strumento professionale collegato via MIDI) e poi ottenerlo stampato (a video o su carta) bello e pronto per essere diffuso.

Come strumenti si possono usare, oltre a quelli forniti, centinaia di altri, disponibili sui dischi di dominio pubblico, oppure, come già detto, strumenti professionali MIDI od anche suoni digitalizzati.

QUANDO IL DRIVE LAMPEGGIA E TUONA

Una routine per accedere al disco operando in linguaggio macchina

di **Walter Adriani**

I due più seccanti difetti nel dialogo tra il C/64 e la sua principale periferica (il drive) sono:

- l'impossibilità di controllare in modo diretto lo stato d'errore;
- il dover stare attenti nell'aprire e chiudere il file di comando, pena l'emissione dei noiosissimi ?FILE OPEN e ?FILE NOT OPEN ERROR.

Il primo inconveniente si presenta quando commettiamo un errore su disco (spia del drive che lampeggia). Non esiste, infatti, un comando per visualizzare lo stato d'errore del drive, e ciò rende ogni errore una tragedia.

Il secondo si presenta ogni volta che vogliamo inviare comandi al drive. Nel caso dimentichiamo se il file di comando è stato già aperto oppure no, verremo multati dagli errori suddetti. Inoltre, se sbagliamo nel digitare un comando, commetteremo un errore, e ricadremo nel primo inconveniente. Ogni dialogo tra computer e periferica diventa così un'avventura in cui, nel migliore dei casi, avremo sprecato tempo, fatica e pazienza nel digitare comandi.

Questo fatto mette in luce l'ennesima carenza del BASIC V2.0 del C/64. Tanto per fare un esempio, nel C/128 esistono istruzioni come COLLECT, HEADER, SCRATCH, ed altre che sostituiscono i comandi del drive. Esiste, poi, la variabile DS$ che informa (Print DS$) l'errore occorso.

Per supplire a questa carenza, piuttosto che emulare il C/128 creando altrettanti comandi, si è pensato di crearne uno che li sostituisca tutti.

La routine proposta, infatti, invia comandi al drive e visualizza, in qualsiasi momento, lo stato d'errore. Quest'ultimo viene automaticamente visualizzato anche dopo l'invio dei comandi, in modo da controllare che tutto si sia svolto correttamente.

La routine può essere utilizzata sia in modo diretto, caricandola con...

```
LOAD "DISK", 8, 1
```

...dopo aver salvato il disassemblato con un Monitor, sia da programma utilizzando il caricatore BASIC pubblicato. Se il comando viene dato da programma, al termine della routine verrà ripresa la sua esecuzione.

DETTAGLI

La routine è allocata a partire da $8000. Le due sintassi sono, rispettivamente...

```
SYS 32820, "comando"
```

...e...

```
SYS 32820, ""
```

Il comando tra virgolette è lo stesso che andava messo nella sintassi PRINT# 15,

"comando" ,ma non dovremo preoccuparci di aprire il file 15.

Ad esempio, per denominare con il nome "PIPPO" il file "PROVA", basterà digitare...

SYS 32820, "R:PIPPO=PROVA"

...Al termine, se tutto si è svolto correttamente, verrà visualizzato il messaggio...
00,OK,00,00
...e comparirà il solito READY.

E' molto comodo trasformare la routine in un comando BASIC aggiuntivo, ad esempio DISK (vedi "Lo Strizzacervelli" su C.C.C.56 e "Comandi nuovi..." su C.C.C. 57). Le nuove sintassi saranno ovviamente DISK "comando" e DISK "" (si può sopprimere il Controllo Virgola JSR $AEFD, come suggerito nei suddetti articoli).

COME FUNZIONA

Il funzionamento della routine è semplice e comprensibile anche ai meno esperti: essa non fa che eseguire in L.M. ciò che prima dovevamo digitare in BASIC.

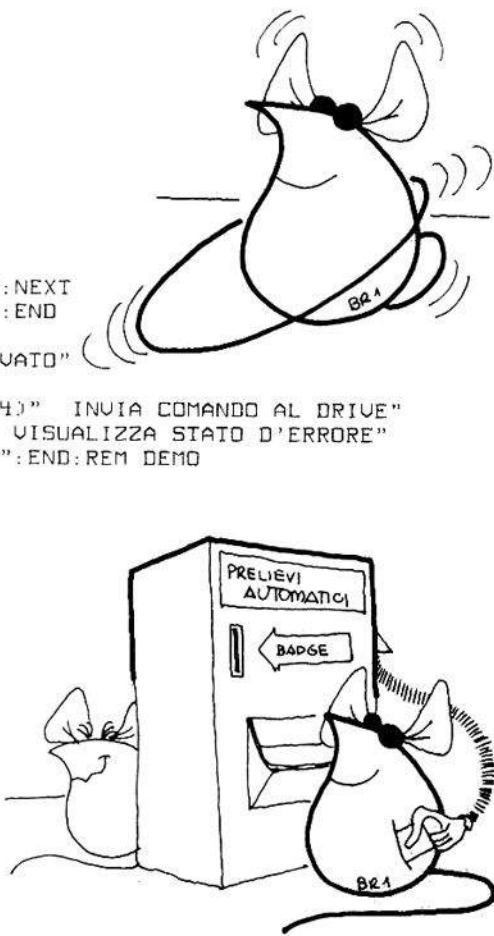
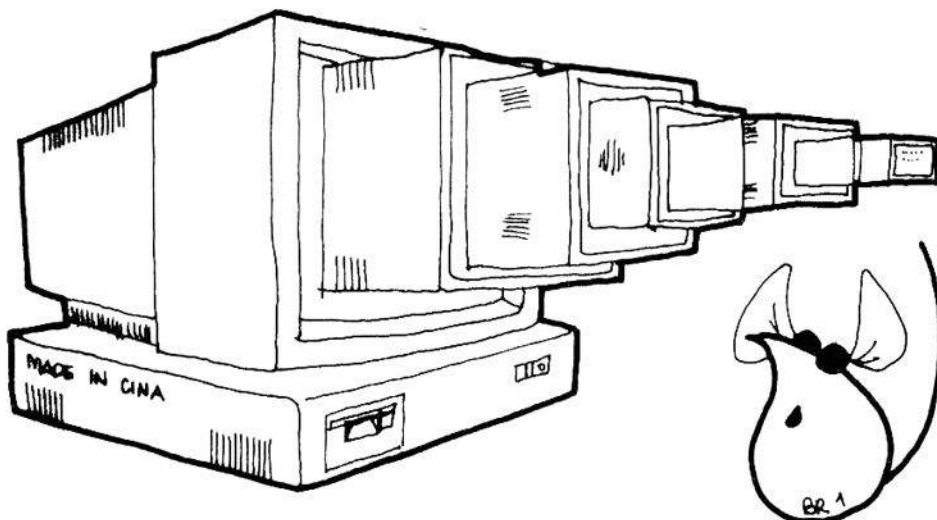
Vengono sfruttate le routine del KERNAL, di cui si è già parlato su C.C.C. 56.

Dapprima viene prelevato il parametro stringa che contiene il comando da inviare. Se la stringa è nulla, si passa direttamente al Controllo Errore, altrimenti si apre il file di comando settando come filename la stringa suddetta: questa operazione invia il comando al drive. Il file di comando viene poi chiuso e si passa al Controllo Errore.

Per controllare lo stato d'errore, si riapre il file di comando, lo si imposta come canale di INPUT (default: tastiera) e si esegue il LOOP $8016 - $8022, che preleva e stampa i caratteri del messaggio d'errore del drive finchè non incontra il codice ASCII di RETURN. Chiuso il file di comando, resettati i canali I/O di default e stampato un ritorno carrello, l'esecuzione termina con un RTS.

Possono essere inviati tutti i comandi che usano il file di comando (comandi di Housekeeping e di Accesso Diretto).

La routine è rilocabile, ma è indispensabile ricordarsi di aggiornare il "top di memoria" (vedi disassemblato) nel caso si desideri spostare il suo inizio. A tale compito provvede automaticamente la routine stessa. La SYS di partenza deve trovarsi, in ogni caso, 52 byte dopo l'indirizzo di partenza.

```
100 REM ******************
105 REM *                *
110 REM *  COMANDO 'DISK' *
115 REM *   BY W.ADRIANI  *
120 REM *  BRACCIANO (RM) *
125 REM *                *
130 REM ******************
135 :
140 FORR=1TO99:READA:POKE32768+R,A:C=C+A:NEXT
150 IFC<>13910THENPRINT"ERRORE NEI DATA":END
155 POKE53280,0:POKE53281,0
160 PRINT""SPC(7)" COMANDO 'DISK' ATTIVATO"
170 PRINTSPC(13)"ISTRUZIONI:"
180 PRINT"SYS32820,"CHR$(34)"..."CHR$(34)"  INVIA COMANDO AL DRIVE"
190 PRINT"SYS32820,"CHR$(34);CHR$(34)"  VISUALIZZA STATO D'ERRORE"
200 PRINT"STATO D'ERRORE:":SYS32820,"":END:REM DEMO
1000 DATA 169,015,162,008,160,015,032
1001 DATA 186,255,169,000,032,189,255
1002 DATA 032,192,255,162,015,032,198
1003 DATA 255,032,207,255,201,013,240
1004 DATA 006,032,210,255,056,176,243
1005 DATA 169,015,032,195,255,032,204
1006 DATA 255,169,141,032,210,255,096
1007 DATA 234,234,234,169,128,133,056
1008 DATA 032,253,174,032,158,173,032
1009 DATA 130,183,152,240,188,132,250
1010 DATA 169,015,162,008,160,015,032
1011 DATA 186,255,165,250,166,034,164
1012 DATA 035,032,189,255,032,192,255
1013 DATA 162,015,032,195,255,056,176
1014 DATA 157
1015 END
```

```
*********************
*                   *
*   Disassemblato   *
*  routine  'DISK'  *
* By Walter Adriani *
*                   *
*********************

Allocata da       : $8000
SYS di partenza   : $8034

-------------------------------------
8000 lda #$0F   Num. File (15)
8002 ldx #$08   Periferica (8)
8004 ldy #$0F   Indirizzo (15)
8006 jsr $ffba  Setta file di comando
8009 lda #$00   Nome = stringa vuota
800b jsr $ffbd  Setta nome
800e jsr $ffc0  Apre file di comando
8011 ldx #$0F   File di comando
8013 jsr $ffc6   in INPUT
8016 jsr $ffcf  Preleva un carattere
8019 cmp #$0d   Carattere = RETURN ?
801b beq $8023  Si: fine LOOP
801d jsr $ffd2  No: stampa carattere
8020 sec         e continua LOOP
8021 bcs $8016   (JMP$8016 rilocabile)
8023 lda #$0F   Chiude file
8025 jsr $ffc3   di comando
8028 jsr $ffcc  Resetta canali I/O
802b lda #$8d   Stampa un ritorno
802d jsr $ffd2   carrello
8030 rts        Torna al BASIC
8031 nop
8032 nop
8033 nop
-------------------------------------
8034 lda #$80   Fissa il 'top'
8036 sta $38     di memoria a $8000
8038 jsr $aefd  Controllo virgola
803b jsr $ad9e  Valuta parametro
803e jsr $b782  Preleva parametro
8041 tya        Se stringa nulla salta
8042 beq $8000   a Controllo Errore
8044 sty $fa    Lungh. stringa in $FA
8046 lda #$0F   Num. file (15)
8048 ldx #$08   Periferica (8)
804a ldy #$0F   Indirizzo (15)
804c jsr $ffba  Setta file di comando
804f lda $fa    Lungh. stringa da $FA
8051 ldx $22    Puntatori alla stringa
8053 ldy $23     (comando da inviare)
8055 jsr $ffbd  Setta comando
8058 jsr $ffc0  Apre file di comando
805b ldx #$0F   Chiude file
805d jsr $ffc3   di comando
8060 sec        Va a Controllo Errore
8061 bcs $8000   (JMP$8000 rilocabile)
-------------------------------------
```

OCCHIO AL SATELLITE!

I satelliti artificiali che nel 1975 giravano sulle nostre teste erano circa 3000, di cui solo 1/10 funzionanti; vediamo perchè ogni tanto cadono

di **Valentino Spataro**

Alcuni mesi fa si è ripresentato il problema dei satelliti che sfuggono al controllo umano e cadono sulle nostre teste. Vi ricordate di quel satellite russo i cui motori utilizzavano energia nucleare? O forse la frase dello speaker televisivo "...è caduto senza causare danni..." senza specificare nè dove nè come vi ha rassicurati a sufficienza?

Probabilmente già sapete perchè i satelliti, di tanto in tanto, precipitano, però l'argomento è "più famoso che conosciuto", come diceva Manzoni. Qualche breve e semplice legge fisica può venire incontro alle nostre irrefrenabili curiosità.

COME PARTE UN RAZZO

Mettere un satellite in orbita non è cosa semplice (viva l'uovo di Colombo): pensate, tanto per cominciare, che lanciare un satellite a est o a ovest non è indifferente: infatti lanciandolo verso est possiamo godere di una velocità iniziale di 450 metri al secondo (se ci troviamo all'equatore) grazie alla rotazione terrestre. Ogni lancio inoltre va generalmente diviso in diverse fasi; considerando che gli stadi di un vettore (il missile che porta il satellite) sono di solito tre, possiamo distinguere 5 fasi:

1) inizio combustione primo stadio e compimento del primo tratto di salita verticale
2) fine combustione primo stadio e inizio secondo (V = 6000 Km/h)
3) fine combustione secondo stadio (V =

15000 Km/h)
4) tratto di volo balistico (senza ulteriore spinta) e inizio combustione terzo stadio (V = 14000 Km/h)
5) fine combustione terzo stadio e separazione del satellite (V = 28000 Km/h)

...e tutto questo, pensate, percorrendo in proiezione 1700 Km di superficie terrestre in soli 10 minuti (in barba al fu decreto Ferri).

Evidentemente questo momento, che è il più importante, è troppo complesso per essere simulato su un computer come il C/64. Possiamo però descrivere quali sono i requisiti necessari perchè un satellite resti in orbita.

UN PO' DI TEORIA

Innanzitutto parliamo di energia totale di un corpo. Cercate nella vostra stanza un blocchetto di marmo (è indispensabile che sia di marmo). Prendetelo in mano e capovolgetelo verso il basso; aprendo la mano che cosa succederà? Escludendo che il blocco di marmo vi sia caduto sul piede, e sperando che non abbia sfondato il pavimento, è successo che l'oggetto è giunto in basso con velocità tanto maggiore quanto più alto era stato il punto di partenza. Ma visto che in Fisica (e non solo lì) nulla si crea e nulla si distrugge questa velocità (o energia) deve provenire da "qualcosa": questo qualcosa, sempre in Fisica, si chiama energia potenziale, vale a dire una quantità di energia non "in atto" ma suscettibile di manifestarsi tramite opportuno lavoro.

Conseguenza del principio che nulla si crea e nulla si distrugge è il principio che afferma che l'energia totale del sistema resta invariata durante l'esperimento. Lasciando libero l'oggetto, l'energia totale si mantiene costante, mentre l'energia potenziale si è "trasformata" in energia cinetica (velocità). Tale energia si è trasferita sulla superficie su cui l'oggetto è caduto diventando calore, vibrazione (o dolore per il vostro piede).

In termini scientifici il fenomeno si indica...

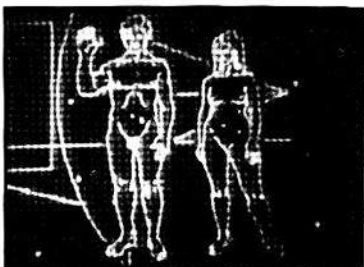
Et = Ep + Ec
(con Et = costante)

...in cui Et è l'energia totale, Ep l'energia potenziale ed Ec l'energia cinetica.

E dire che fino a oggi non vi eravate mai preoccupati di una biro che cadeva in terra!

Lo stesso avviene nei satelliti: una volta messi in orbita (l'oggetto preso in mano) essi dispongono di una ben determinata quantità di energia totale che non può essere variata se non da motori ausiliari (che corrisponderebbero al lavoro compiuto dalla nostra mano per spostare in altezza un oggetto).

Se vogliamo posizionare un satellite su un'orbita più alta (e vi resti a lungo) dobbiamo essere disposti a pagare un prezzo, che consiste nella minore velocità di rotazione del satellite attorno alla Terra.

L'intero procedimento è facilmente quantificabile se il sistema che ci accingiamo a studiare è composto dal solo pianeta. Ma visto che, generalmente, i pianeti non si trovano come i limoni al mercato (ma in luoghi più ampi dal nome di "sistemi solari" comprendenti, a loro volta, altri pianeti e relative lune e con, generalmente, un sole in mezzo) i conti non sono mai così facili. E' peraltro vero che se il nostro satellite monopolizzato dalla RAI deve solo trasmettere i programmi stranieri in Italia è evidente che deve restare più vicino alla Terra che non ad altri corpi celesti, lasciando gli influssi degli altri pianeti sufficientemente irrilevanti (però non si sa mai!).

VELOCITA' E ORBITA DEI SATELLITI

La forma dell'orbita di un satellite artificiale dipende, strano ma vero, dalla velocità dello stesso. Innanzitutto definiamo come velocità circolare (Vc) la velocità che un satellite deve avere (in corrispondenza di una determinata altezza H dalla superficie di un pianeta di raggio R) per mantenere un'orbita circolare (prima di proseguire rileggere e imparare a memoria).

Credete forse di poter continuare a leggere impunemente senza aver riletto e imparato a memoria? Ho capito: continuiamo.

Per calcolare Vc si deve usare la seguente formula:

Vc = G * M / (R + H)

dove G è la costante di gravitazione universale o di Cavendish, M e R, rispettivamente, massa e raggio del pianeta che si considera, H la distanza del satellite dalla superficie (la massa del satellite è considerata unitaria).

Definendo "V" la velocità del satellite nel punto P in cui il satellite si sgancia dal razzo vettore, varranno le seguenti relazioni:

1) se V < Vc allora l'orbita è ellittica con P apogeo dell'ellisse
2) se V = Vc allora l'orbita è circolare
3) se V > Vc e V < Vc * sqr(2) allora l'orbita è ellittica e P è il perigeo dell'ellisse
4) se V = Vc * sqr(2) allora l'orbita è parabolica
5) se V > Vc * sqr(2) allora l'orbita è iperbolica

Ricordiamo che in un'orbita ellittica il perigeo è il punto più vicino al pianeta e l'apogeo il più lontano.

Nel primo caso c'è da dire che l'orbita può passare anche "attraverso" il pianeta; naturalmente l'impatto è in questo caso assicurato.

E' anche vero che basta attraversare l'atmosfera terrestre perchè gli attriti, assenti nello spazio, tornino a farsi vivi per complicare la vita a noi poveri mortali che, disgraziatamente, siamo interessati a queste cose.

E' infine evidente che nel caso 4) e 5) il satellite è destinato ad allontanarsi definitivamente dal nostro amato pianeta (amato, sì; ma da chi?).

LE FORMULE

Assegnata la distanza del satellite dalla superficie terrestre, il programma di queste pagine calcola la velocità circolare (Vc) e Vc * sqr(2), cioè la velocità necessaria perchè il satellite si allontani definitivamente dalla Terra (casi 4 e 5 ut supra); calcola anche il tempo che il satellite impiega a percorrere l'orbita (periodo) e la lunghezza della stessa.

Considerando che un satellite, posizionato all'altezza di 150 Km, "vive" solo due giorni (per via degli attriti con l'atmosfera) e che con altezze superiori a 3000 Km può vivere per migliaia di anni, provate a fornire i seguenti valori: 200, 1800, 10000 e 36000 Km. Con quest'ultimo dato vedrete che il periodo di rotazione è di circa 24 ore: tale satellite è dunque geostazionario, il che significa che, ruotando nello stesso periodo della Terra, resta fermo sempre sulle medesime coordinate. Sapendo che per velocità di allontanamento si intende Vc * sqr(2) potrete interpretare i risultati con lo schema proposto nel paragrafo precedente.

Riefenamoci ora ad un esempio concreto: inserite una distanza dalla superficie di 36000 chilometri; avrete una Vc di 11041

Km/h e Vc * sqr(2) uguale a 15615 Km/h circa. Se quindi un satellite ha velocità V nel punto P (dove il satellite si sgancia dal razzo vettore) uguale o maggiore di Vc certamente la sua orbita non correrà il rischio di avvicinarsi troppo alla Terra. In particolare, se V = Vc il satellite resterà sempre alla stessa distanza; se V > Vc e V < Vc * sqr(2) il satellite si trova già nella posizione più vicina alla Terra (perigeo); se infine V => Vc * sqr(2) il satellite uscirà dall'attrazione terrestre.

Con V < Vc il satellite può continuare a tenere l'orbita solo se essa non passa per l'atmosfera (se non, addirittura, attraverso il pianeta); per saperlo è necessario calcolare, di volta in volta, l'orbita e verificare.

E' proposta anche una applicazione (tracciamento di grafico: in X distanze e in Y la Vc) che deve essere considerata solo come spunto per ulteriori approfondimenti.

Se il programma potrà suggerirvi la creazione di qualche gioco, ricordatevi, per amor del cielo, che le unità di misura vanno rispettate. Il programma usa il sistema MKS, ed è comunque ampiamente commentato per essere compreso con facilità.

I LISTATI

E' presente anche una versione per personal computer in Basic. Può essere una occasione utile per mettere a confronto due menti diverse.

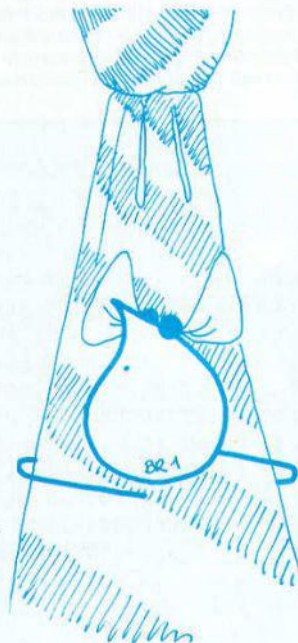
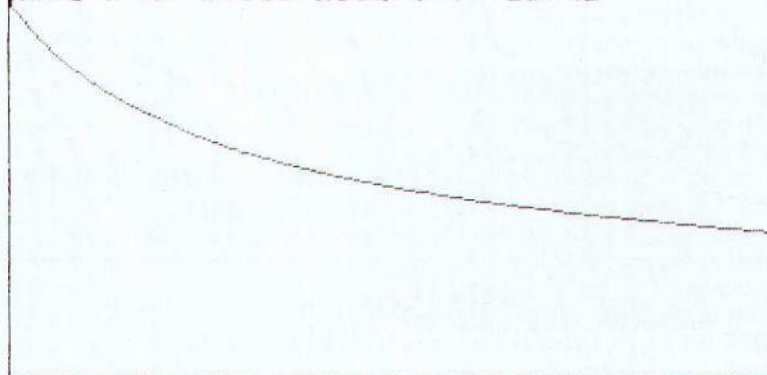
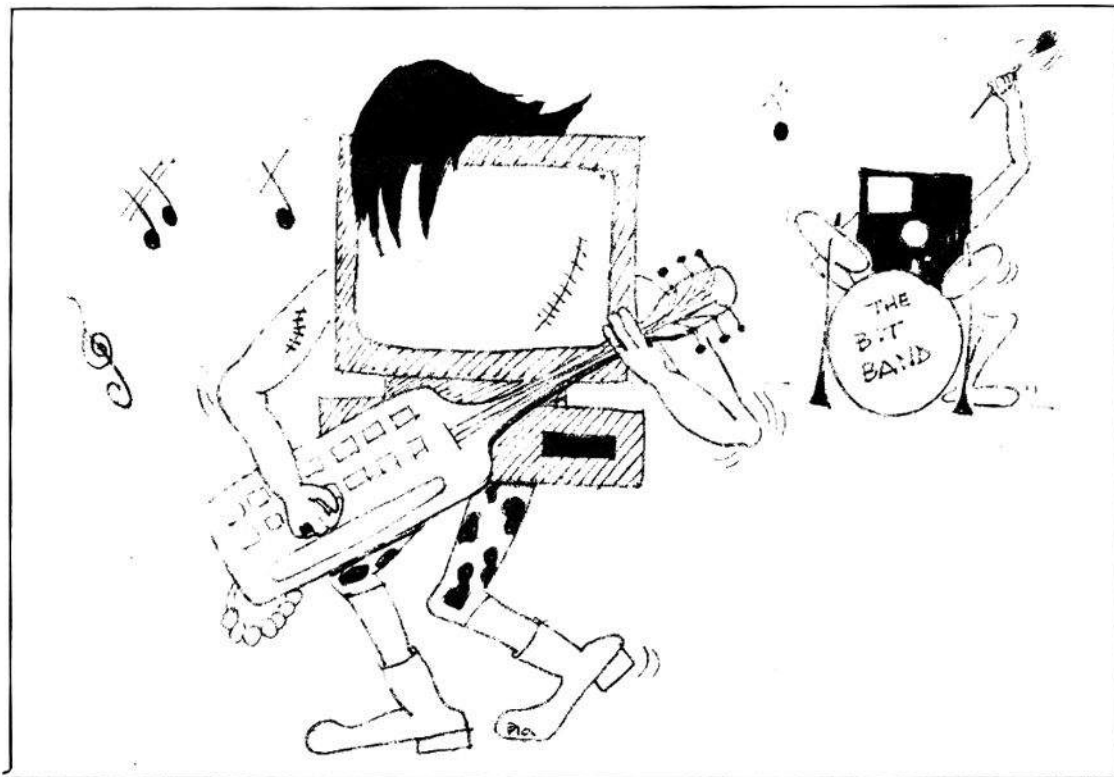
Come potrete notare con un colpo d'occhio i due programmi sono praticamente identici (l'istruzioni PRINT CHR$(147) per C/64 diventa CLS; PRINT CHR$(19) diventa LOCATE 1,1). Unica diversità fondamentale è nelle variabili. Nel C/64 sappiamo (si spera per voi) dell'esistenza di tre tipi di variabili: stringa (A$), intere (A%) e reali (A). Nei PC, oltre a queste variabili, ve ne sono di altri tipi. Uno di questi è contrassegnato dal carattere cancelletto (#) che sta a indicare una variabile in doppia precisione (esempio: A#). Per chi usa il C/64 questa è una chimera data la notoria imprecisione dei calcoli del nostro veterano calcolatore. Di fronte alla necessità di gestire, con i PC, conti a nove cifre senza sbagliare (altrimenti i conti non tornano), sono state create tali variabili a vantaggio di chi ha conti correnti a nove cifre (o più), o di chi si diletta, più modestamente, di astronomia con annessi e connessi.

Un'ultima avvertenza: i risultati forniti dal computer hanno una percentuale minima di errore, dovuta all'imprecisione dei dati relativi alla Terra (massa e raggio): se pensate ad applicazioni professionali è consigliabile recarsi presso il più vicino laboratorio astronomico, ente spaziale o centro di spionaggio e chiedere i relativi dati.

```
100 REM CALCOLO VELOCITA' ORBITALE DI UN SATELLITE
110 REM BY V.SPATARO PER PC + GW-BASIC. USATA UNITA' MKS DI MISURA
115 FOR P=1 TO 20: K$=K$+CHR$(32):NEXT
120 G#=6.67*10^(-11):REM COSTANTE GRAVITAZIONALE IN N*M^2/KG^2
130 MT#=5.977*10^24:REM MASSA DELLA TERRA IN CHILOGRAMMI
140 R#=6378388!:CLS:REM RAGGIO TERRESTRE IN METRI
150 LOCATE 1,1:PRINT"ALTEZZA DA SUPERFICIE TERRESTRE IN KM ":INPUT H1:H#=H1*1000
160 GOSUB 210:REM CALCOLI IN BASE AD H (ALTEZZA) ESPRESSA IN METRI
170 PRINT:PRINT"PERIODO (HH:MM)"INT(P#/60)":"INT(P#-INT(P#/60)*60);K$
180 PRINT"ALTEZZA DA SUPERFIC.(KM)"H1;K$
185 PRINT"VELOCITA' CIRCOLARE  (KM/H)"INT(V#);K$
190 PRINT"VELOC. DI ALLONTANAM.(KM/H)"INT(V#*SQR(2));K$
200 PRINT"LUNGHEZZA ORBITA IN KM ";INT(L#/1000);K$:PRINT:GOTO 150
210 REM SUBROUTINE CALCOLI
220 V#=SQR((G#*MT#)/(R#+H#))*3.6:L#=(R#+H#)*2*3.14:P#=L#/(V#*100/6):RETURN
230 REM V IN KM/H; L (ORBITA) IN METRI; P (PERIODO) IN MINUTI
240 END
```

```
100 REM diagramma cartesiano relativo a calcolo velocita' satellite
110 REM         by valentino spataro    versione Gw-Basic
111 REM          idonea per computer Ms-Dos compatibili
113 :
120 G=6.67*10^-11:MT=5.977*10^24:R=6378388!:PI=3.14159265#
130 SCREEN 1:CLS 1:COLOR 7,0,0:LINE (0,0)-(0,199):LINE -(329,199):VX=0:VY=199
140 MIN=200000!:MAX=36000000#:VMAX=30000:PASSO=MIN
150 FOR H=MIN TO MAX STEP PASSO:GOSUB 200:RX=H/(MAX/320):RY=V/(VMAX/200)
160 LINE (VX,200-VY)-(RX,200-RY):VX=RX:VY=RY
170 LOCATE 1,1:PRINT"asse x:h="+STR$(INT(H/1000))+" asse y:v="+STR$(INT(V)):NEXT
180 A$=INKEY$:IF A$="" GOTO 180:ELSE SCREEN 0:END
190 REM subroutine calcoli. usato sistema mks
200 V=SQR((G*MT)/(R+H))*3600/1000:L=(R+H)*2*PI:P=L/(V*100/6):RETURN
210 REM v=v*3600/1000 converte v da m/s a km/h
220 END
```

```
100 REM DIAGRAMMA CARTESIANO APPLICATIVO A CALCOLO VELOCITA' SATELLITE
110 REM BY SPATARO VALENTINO. PRG PER C64+ GWBASIC V2.0
120 G=6.67*10↑-11:MT=5.977*10↑24:R=6378388
130 SCREEN 1:CLS 1:COLOR 7,0,0:LINE (0,0)-(0,199):LINE -(329,199):VX=0:VY=199
140 MIN=200000:MAX=36000000:VMAX=30000:PASSO=MIN
150 FOR H=MIN TO MAX STEP PASSO:GOSUB 200:RX=H/(MAX/320):RY=V/(VMAX/200)
160 LINE (VX,200-VY)-(RX,200-RY):VX=RX:VY=RY
170 LOCATE 1,1:PRINT"ASSE X:H="+STR$(INT(H/1000))+" ASSE Y:V="+STR$(INT(V)):NEXT
180 GET A$:IF A$="" GOTO 180:ELSE SCREEN 0:END
190 REM SUBROUTINE CALCOLI. USATO SISTEMA MKS
200 V=SQR((G*MT)/(R+H))*3600/1000:L=(R+H)*2*π:P=L/(V*100/6):RETURN
210 REM V=V*3600/1000 CONVERTE V DA M/S A KM/H
220 END
```

ASSE X:H= 36000 ASSE Y:V= 11041

```
100 REM CALCOLO VELOCITA' ORBITALE DI UN SATELLITE
110 REM VERSIONE C/64 + EMULATORE GW-BASIC
120 G=6.67*10↑-11:REM COSTANTE GRAVITAZIONALE IN N*M↑2/KG↑2
130 MT=5.977*10↑24:REM MASSA DELLA TERRA IN CHILOGRAMMI
140 R=6378388:PRINTCHR$(147):REM RAGGIO TERRESTRE IN METRI
150 PRINT CHR$(19)"ALTEZZA DA SUPERFICIE TERRESTRE IN KM ":INPUT H1:H=H1*1000
160 GOSUB 210:REM CALCOLI IN BASE AD H (ALTEZZA) ESPRESSA IN METRI
170 PRINT:PRINT"PERIODO (HH:MM)"INT(P/60)":"INT(P-INT(P/60)*60)
180 PRINT"ALTEZZA DA SUPERFICIE (KM)"H1:PRINT"VELOCITA' CIRCOLARE  (KM/H)"INT(V)
190 PRINT"VELOC. DI ALLONTANAM.(KM/H)"INT(V*SQR(2))
200 PRINT"LUNGHEZZA ORBITA IN KM ";INT(L/1000):PRINT:GOTO150
210 REM SUBROUTINE CALCOLI
220 V=SQR((G*MT)/(R+H))*3.600:L=(R+H)*2*π:P=L/(V*100/6):RETURN
230 REM V IN KM/H; L (ORBITA) IN METRI; P (PERIODO) IN MINUTI
240 END
```

TAPE LISTEN II

Finalmente la versione che funziona sul C/128 in modo 64 oltre che sul 64 "normale"; ed anche una sorpresa per i C/16-sti

di **Lorenzo Emilitri**

Mea culpa, mea culpa, mea grandissima culpa. Ebbene si, questa volta abbiamo proprio sbagliato.

Ci riferiamo, naturalmente, al programma presentato nell'articolo "Un insolito mangia-cassette musicale" che, pur funzionando perfettamente sul C/64, non portava ai risultati voluti sul C/128 attivato in modo 64.

Il motivo del mancato funzionamento era da ricondursi alla non totale compatibilità del C/128, sulla quale a volte ci siamo intrattenuti. A nostra discolpa possiamo solo dire di aver verificato il vecchio listato solo su un C/64 d'annata (1984).

Il listato presentato su queste pagine, invece, funziona sul C/128 normale, sul C/128-D (attivati, lo ripetiamo, in modo 64) ed anche sul C/64 "normale", non contenuto, quindi, nel fratello maggiore 128.

ED ANCHE PER C/16 E PLUS-4!

E veniamo ora alla più grossa novità, ovvero alla versione C-16/+4. Essa non è altro che un adattamento della versione 64 sulle due macchine, pur se con qualche modifica causata dalle diversità presenti nell'hardware.

I PROGRAMMI

A seconda che possediate un C-64/128 ó un C-16/+4, digitate la versione idonea, salvatela su nastro o disco e mandatela in esecuzione.

Nella versione C-16/+4 verrà chiesto esplicitamente di premere il tasto PLAY sul registratore, mentre in quella 64/128 no-

UN ULTERIORE CONTRIBUTO

Il lettore Claudio Verniani di Bologna ha inviato una versione per C/16 leggermente diversa da quella proposta da Emilitri. La pubblichiamo egualmente, per completezza, riportando anche alcune note esplicative che saranno apprezzate da chi non possiede il fascicolo citato di Commodore Computer Club.

Il Plus/4 ed il C/16 sono macchine dotate di un generatore di suono non paragonabile al corrispondente chip del C/64, ma sufficiente per molti scopi.

Il programma di queste pagine, molto semplice da usare e da comprendere, ne utilizza le prerogative.

Dopo aver digitato il programma, e averlo salvato su disco o nastro, inserite nel registratore il nastro di musica rock preferito, premete il tasto play, e digitate il comando RUN.

Non gridate al miracolo se la musica della vostra cassetta preferita viene riprodotta "digitalizzata" dal computer: la spiegazione del fenomeno, infatti, non è molto complessa.

Tutti sanno che il registratore a cassette non è stato realizzato per riprodurre musica, ma progettato per memorizzare e fornire dati digitali; è quindi in grado di "riconoscere" solo due livelli logici (alto e basso). Se inseriamo un nastro musicale nel registratore commodore, questo lo tratterà come una normale cassetta di dati e programmi, inviando al computer il suo contenuto.

Dal momento che il datassette è in grado di riconoscere solo segnali digitali, e che in un nastro musicale esistono infiniti livelli intermedi, si potrebbe pensare che sia impossibile udire qualcosa. La musica, ovviamente, risulterà distorta, ma "riconoscibile".

E' sufficiente prelevare, dal bit 4 del registro di memoria $0001, il bit in cui è depositato il segnale del nastro e portare tale valore al sintetizzatore musicale dei computer della serie Ted (C/16/116/+4). Ricordo agli inesperti che i registri del ted inerenti ai generatori sonori sono allocati nei byte $FF0E, $FF0F, $FF10, $FF11.

terete che il bordo cambia colore (da nero diverrà giallo o grigio).

In entrambi i casi, inserite la vostra cassetta preferita nel Datassette e premete PLAY. Non aspettatevi un'elevata qualità di riproduzione sonora; questa è tuttavia sufficiente per capire chiaramente il brano musicale presente su nastro.

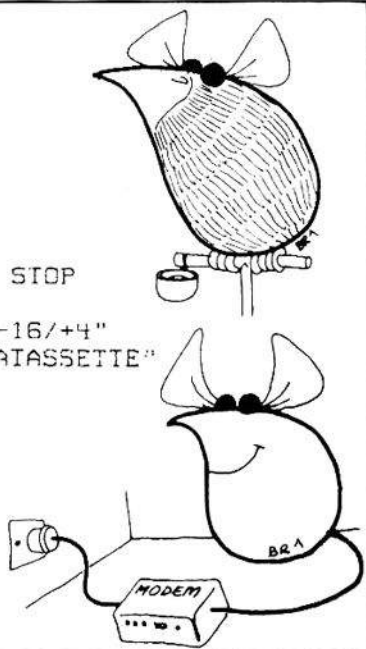
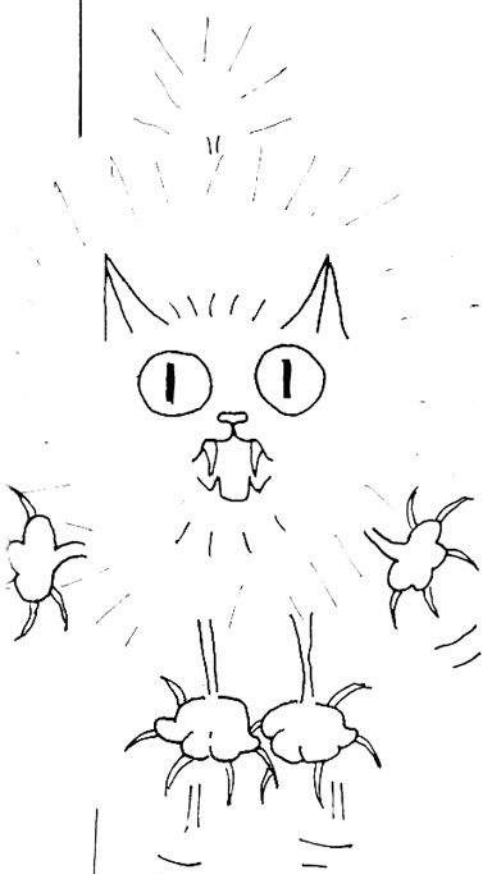
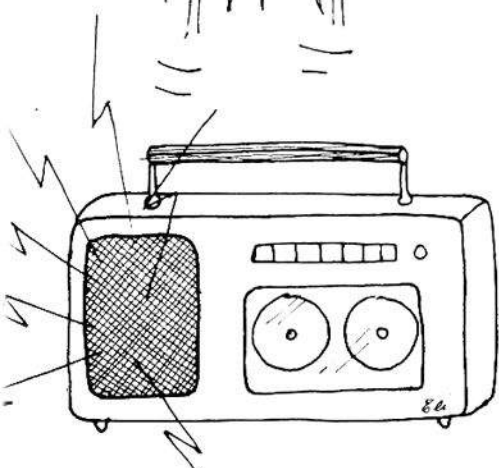
Gli effetti migliori si ottengono con la musica House (come sarebbe a dire "Cosa diavolo è?"? Non diteci di non aver mai ascoltato almeno una volta "Pump up the volume" o "Theme from S-Express"!).

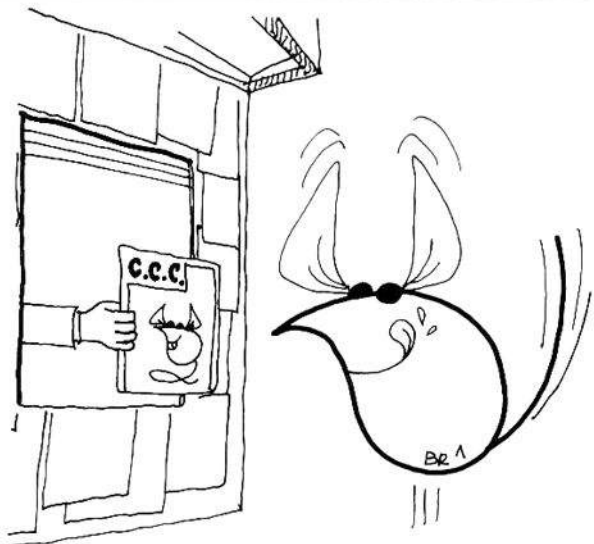
SVILUPPI, SPERANZE & PREVISIONI

Con un po' di fantasia, una basilare conoscenza d'elettronica e le attrezzature adatte, si potrebbe utilizzare il segnale sonoro letto dal computer per realizzare dei pezzi House autocostruiti... oppure, con una tecnica simile, inserire il segnale proveniente dall'elaboratore in propri brani musicali (sconsiglio vivamente, però, di accostarli a Mozart o Beethoven...).

Il limite, come sempre, è dato dalla vostra fantasia!

(Ringraziamo l'amico Federico Baranzini per la gentile collaborazione alla versione 128 e la gatta Zarina per il sostegno morale durante la stesura del presente articolo.)

```
100 REM * * * * * * * * * * * * * * * *
110 REM * --- TAPE LISTEN  C16/+4 --- *
120 REM * BY  LORENZO EMILITRI        *
130 REM * * * * * * * * * * * * * * * *
140 :
150 FOR A=0 TO 33
160 : READ B : C=C+B
170 : POKE 1536+A,B
180 NEXT A
190 IF C<>4871 THEN PRINT"ERRORE NEI DATA": STOP
200 COLOR 0,1: COLOR 4,1: COLOR1,6
210 PRINT CHR$(147) TAB(10) "TAPE LISTEN  C-16/+4"
220 PRINT TAB(3) "INSERISCI UN NASTRO NEL DATASSETTE"
230 SYS 1536
240 :
250 REM DATA PER LINGUAGGIO MACCHINA
260 REM VEDI DISASSEMBLATO COMMENTATO
270 DATA 032,027,227,165,001,041,016
280 DATA 208,021,169,240,141,017,255
290 DATA 141,025,255,169,002,141,008
300 DATA 255,173,008,255,201,255,240
310 DATA 230,096,169,247,208,233
320 END
```

```
;* * * * * * * * * * * * * * * * * * * *
;*     tape listen 128 (modo 64)       *
;*        by lorenzo emilitri          *
;* * * * * * * * * * * * * * * * * * * *
;
      *=$c000    ;inizio a 49152
;
      sei         ;disab. interrupts.
      lda $00     ;abilita il bit 3
      and #$f7    ;della porta come
      sta $00     ;ingresso.
      lda $01     ;accendi il motore
      and #$df    ;del registratore.
      sta $01     ;
loop  lda $01     ;carica val. $01
      sta video   ;mettilo nel video
      sta volume;e nel volume.
      lda joy2    ;se il joy (2) non
      cmp #$7f    ;e' stato mosso
      beq loop    ;salta a loop.
      cli         ;se no riab. irq
      rts         ;ed esce.
; simboli:
video  equ $d020
volume equ $d418
joy2   equ $dc00
```

```
;  ***  Disassemblato Tape Listen  ***
;  ***  Versione Commodore 16  +4  ***
;  ***       By Lorenzo Emilitri   ***
;
. 2000   jsr $e31b ;Stampa 'PRESS...
. 2003   lda $01   ;Carica val. loc. 1
. 2005   and #$10  ;Azzera gli altri bit
. 2007   bne $201e ;Se valore alto va
                   ;a $201e
. 2009   lda #$f0  ;Mette valore basso
. 200b   sta $ff11 ;nel volume
. 200e   sta $ff19 ;e nel bordo.
. 2011   lda #$02  ;Se c'e' un tasto
. 2013   sta $ff08 ;premuto...
. 2016   lda $ff08 ;
. 2019   cmp #$ff  ;
. 201b   beq $2003 ;...esce
. 201d   rts       ;(Torna al BASIC)
. 201e   lda #$f7  ;Mette val. alto
. 2020   bne $200b ;e va a $200b
```

```
100 REM * * * * * * * * * * * * * * * * * *
110 REM *     TAPE LISTEN  C-64/128  - *
120 REM *      BY LORENZO CHILITRI      *
130 REM * * * * * * * * * * * * * * * * * *
140 :
150 REM CARICAMENTO ROUTINE L/M
160 FOR A = 0 TO 29
170 : READ B : C=C+B
180 : POKE 49152+A,B
190 NEXT A
200 IF C<>3580 THEN PRINT "ERRORE": STOP
210 POKE 53280,0 : POKE 53281,0
220 PRINT CHR$(147) CHR$(30)
230 PRINT TAB(10) "TAPE LISTEN C-64/128"
240 PRINTTAB(3) "INSERISCI UN NASTRO"
250 SYS 49152
260 POKE 53280,0
270 END : END : END : END : END : END
280 :
290 REM DATA PER LINGUAGGIO MACCHINA
300 REM VEDI DISASSEMBLATO COMMENTATO
310 DATA 120,165,000,041,247,133,000
320 DATA 165,001,041,223,133,001,165
330 DATA 001,141,032,208,141,024,212
340 DATA 173,000,220,201,127,240,241
350 DATA 088,096
360 END
```

```
10 FORI=DEC("1300")TODEC("1300")+71
20 READZ$:POKEI,DEC(Z$):Q=Q+DEC(Z$)
30 NEXT
40 IFQ<>7061THENPRINT"ERRORE NEI DATA":END
41 PRINT"PREMI UN TASTO SUL REGISTRATORE":DO:LOOP WHILE PEEK(64784)AND4
42 PRINT"  MUSIC TAPE C-16 PLUS.4 VERSION
43 PRINT"INSERIRE NASTRO  DI MUSICA E PREMI PLAY"
44 PRINT"           BY VERNIANI CLAUDIO 1988"
47 :
50 SYSDEC("1300")
1300 DATAA9,40,8D,19,FF,EA,EA,EA
1308 DATA78,A5,01,29,10,C5,10,D0
1310 DATA1F,A9,00,8D,11,FF,A5,46
1318 DATA8D,0F,FF,4C,09,13,CE,15
1320 DATAFF,4C,29,10,00,00,00,00
1328 DATA00,00,00,00,00,00,00,00
1330 DATAA9,FF,8D,11,FF,A9,00,8D
1338 DATA0F,FF,8D,0E,FF,EE,15,FF
1340 DATAEE,19,FF,4C,09,13,00,00
1350 END
```

DISASSEMBLATO DEL PROGRAMMA
MUSIC TAPE 88
BY VERNIANI CLAUDIO VRN SOFT 88

```
INDIRIZZI
   STRT  $1300  DEC= 4864
   SUB   $1330
   PROG  $1309

STRT LDA #$ 40  INIZIALIZZA
     STA $FF19  COLORE SCHERMO
     SEI        DIS. INTERRUPTS

PROG LDA $01    CARICA SEGNALE NASTRO
     AND #$10   PRENDE IL BIT 4
     CMP $10    VERIFICA SEGNALE
     BNE $SUB   SE C'E FAI SUB
     LDA #$00   AZZERA  LOCAZIONE
     STA $FF11  VOLUME
     LDA $46    CAMBIA FREQUENZA
     STA $FF0F  VOCE 2
     JMP $PROG  RIPETI

SUB  LDA #$FF   ALZA
     STA $FF11  LOC. VOLUME
     LDA #$00   AZZERA
     STA $FF0F  FREQUENZA 2
     STA $FF0E  FREQUENZA 1
     INC $FF15  INC. SCHERMO COLORE
     INC $FF19  INC. BORDO COLORE
     JMP $PROG  RIPETI
```

CAMPUS

LABORATORIO SOFTWARE DI COMMODORE COMPUTER CLUB

ESORDIENTI

- La parola del principiante
- MS-DOS, chi era costui

ESPERTI

- Faccia a faccia con L'HI-RES

CAMPIONI

- Uno scrolling in scatola di montaggio
- Un singolare inseguimento

PROTEZIONI

- Un anti-virus per il tuo drive

LA PARABOLA DEL PRINCIPIANTE

Che cosa bisogna fare per risolvere, con il nostro personal computer, un determinato problema

di **Alessandro de Simone**

(Quasi) qualsiasi computer è in grado di accettare i programmi di queste pagine

Quando si parla di computer si pensa, molto spesso, a prodigiose macchine in grado di risolvere qualsiasi problema di qualsivoglia natura.

Alcuni film di pseudo fantascienza, purtroppo, (per non parlare degli orrendi cartoons giapponesi) alimentano l'illusione per cui sembrerebbe sufficiente "parlare" alla macchina per veder visualizzati grafici, tabelle e, in una sola parola, la soluzione del problema.

La realtà, come avrà avuto modo di accorgersi anche il più distratto dei principianti, è certamente diversa per molteplici difficoltà che sono sempre in agguato per renderci la vita irta di pericoli.

In questo articolo, pertanto, vedremo che cosa può capitare a chi voglia trasferire, su computer, un certo argomento per affrontarlo, per quanto possibile, in un modo "familire" e, soprattutto, utile.

Dal momento che si parlerà di applicazioni grafico-matematiche è risultato indispensabile utilizzare macchine in grado di riprodurre grafici e schermate in alta risoluzione.

I listati presenti in queste pagine, pertanto, sono idonei a girare su computer professionali Ms-Dos (dotati di linguaggio interprete Gw-Basic o compatibile), sul C/128, sul C/16, sul Plus 4 e sul C/64 dotato, però, dell'emulatore Gw-Basic, che la Systems editoriale propone su dischetto per corrispondenza.

Il C/64 da solo, infatti, non consente un'agevole gestione delle pagine hi-res ed è questo il motivo per cui non viene presa in esame una versione C/64 che non tenga conto dell'utilizzo di un Tool grafico specifico.

IL PROBLEMA

L'inserto del N. 47 ("Rette nello spazio") affrontava in modo sistematico lo studio delle coordinate ed il corrispondente "adattamento" ad un personal computer.

Per non ripetere cose già dette, ci limiteremo a ricordare che (vedi figura 1) nello studio di un qualsiasi problema di matematica o geometria siamo abituati a considerare il piano cartesiano con l'origine ("O") posta nell'intersezione degli assi X ed Y. Per ciò che riguarda il segno, poi, consideriamo positive le ascisse poste a destra dell'origine e negative quelle a sinistra; le ordinate sono considerate positive se posizionate al di sopra e negative se al di sotto del punto "O".

Per un personal computer, invece, l'origine è posta nell'angolo in alto a sinistra del monitor; l'asse X è positivo verso destra (come nella "nostra" convenzione) ma l'asse Y risulta positivo (e non negativo) verso il basso.

Tale diversità di vedute costringe ad introdurre una elaborazione specifica per fare in modo che, sullo schermo, appaia ciò che realmente ci si aspetta.

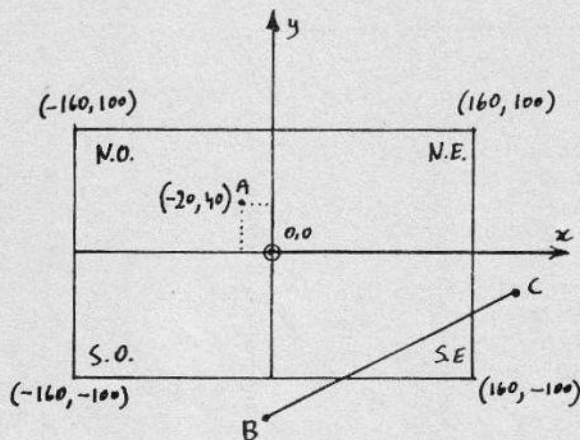
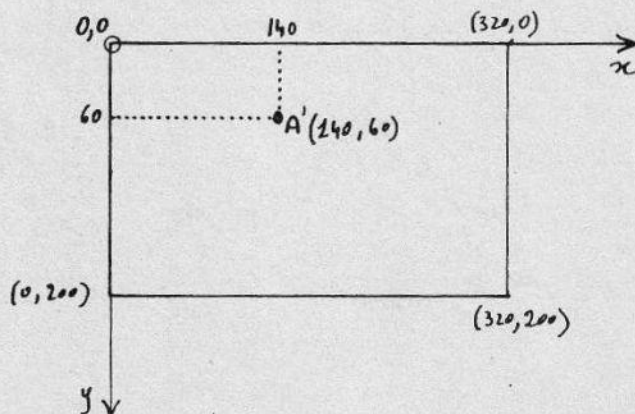
SEGMENTI

I primi due listati ("Segmenti"), riportati nella versione, rispettivamente, Gw-Basic e C/128, consentono di visualizzare un qualsiasi segmento di cui si conoscono le coordinate degli estremi (vedi righe 170 e 180). Il programma, infatti, traccia dapprima una cornice (righe 200 - 230) e poi gli assi di riferimento (240 - 250) in modo da "orientarsi" sul piano cartesiano.

LE AVVENTURE DI

PRIMO GIOVEDINI

by Marco Mietta

"Primo - Emulator"

1

Sembrerebbe, quindi, che il problema della rappresentazione dei segmenti sia risolto. In effetti vi sono casi particolari sui quali invitiamo il lettore a riflettere:

• Nei casi in cui uno dei due punti estremi si trovi al di fuori del campo visibile, possono verificarsi alcuni inconvenienti: mancata visualizzazione oppure tracciamento di segmenti diversi da ciò che si desidera.
• Emissione di messaggi di errore (Illegal quantity error) nel caso in cui una delle due coordinate, o entrambe, superi il limite del range disponibile.

I due tipi di computer (Ms-Dos e C/128), infatti, accettano limiti diversi: il Gw-Basic elabora valori di coordinate compresi tra -32768 e +32767; il C/128, invece, rifiuta valori negativi ed emette il corrispondente messaggio di errore nel caso si veda costretto a trattare numeri non positivi.
E' bene, pertanto, che il lettore si soffermi su questo particolare aspetto del problema, soprattutto nei casi in cui voglia adattare, al proprio computer, listati scritti per girare su altri elaboratori o supportati da determinati linguaggi. Non è affatto sufficiente, in altre parole, trasformare il comando Line (Gw-Basic) nel corrispondente Draw (C/128), o viceversa, per veder girare correttamente un certo programma; è quasi sempre necessario tener conto di altri fattori, tra cui, appunto, il differente range accettato dalle varie macchine, la diversa impostazione dei colori, il codice dei comandi e così via.

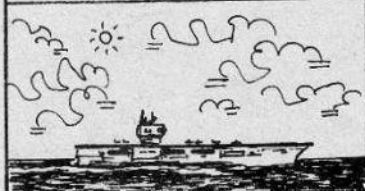
Suggerimenti

Sia ai possessori di C/128 sia a quelli che usano elaboratori Ms-Dos suggeriamo di:

• Introdurre un "filtro" software che impedisca di accettare valori delle coordinate esterni al range consentito. Esempio:
...If X1 < -32768 or X1 > 32767 Then...
...If X1 < 0 Then Print "errore"...

• Aggiungere un'opzione che consenta di prendere in considerazione un solo quadrante alla volta, in modo da ampliare lo spazio su cui operare.

• Prevedere una funzione "Zoom" che consenta di visualizzare segmenti i cui estremi superino di gran lunga i limiti consentiti oppure di lunghezza troppo piccola per esaminarli con cura.

E' già passato un mese da quando, nello scorso episodio, Primo Giovedini debellò i "pirati" di software, con una mossa degna del migliore Interrupt... Ora sembra essere ritornata la calma: i dischetti si formattano a 664 blocchi...

... e Giovedini si concede una pausa dal suo lavoro...!

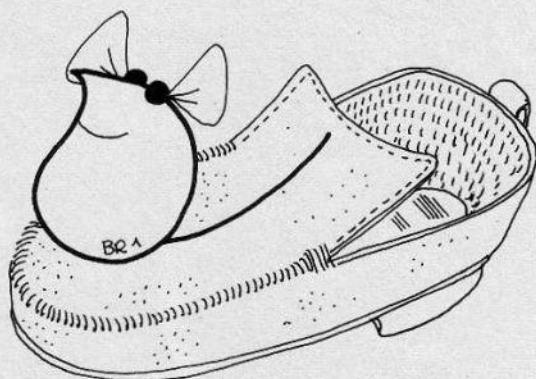
Sys, fortuna che questo video-game ha l'opzione "PAUSA"...

```
100 REM PARABOLE (VERSIONE C/128)
110 :
150 Q=147:W=19
160 PRINT CHR$(Q):REM CANCELLA SCHERMO
170 PRINT CHR$(W):REM HOME
180 INPUT "COEFFICIENTE A";A
190 INPUT "COEFFICIENTE B";B
200 INPUT "COSTANTE     C";C
210 INPUT "STEP         D";D:IF D=0 THEN D=5
220 D$="S":INPUT "CLEAR SCREEN (S/N)";D$
230 GRAPHIC 1:IF D$="S" THEN GRAPHIC 1,1
240 DRAW 1,0,100 TO 319,100:REM ASSE X
250 DRAW 1,160,0 TO 160,199:REM ASSE Y
260 FOR X1=-160 TO 160-D  STEP D
270 Y1= A*X1*X1 + B*X1 + C:REM FORMULA
280 Y2= A*(X1+D)*(X1+D)+B*(X1+D)+C
282 REM IL C/128 NON ACCETTA VALORI DI
283 REM COORDINATE NEGATIVI (CFR. RIGA 285)
285 IF Y1>100 OR Y2>100 THEN 300
290 DRAW 1,160+X1,100-Y1 TO 160+X1+D,100-Y2
300 NEXT
310 GET A$:IF A$="" THEN 310
320 IF A$="." THEN GRAPHIC 0:END
330 GRAPHIC 0:GOTO 170
```

Ai possessori di computer Ms-Dos consigliamo di:

- Introdurre le modifiche del caso per sfruttare per intero lo schermo. Nel caso di 80 colonne, infatti, l'ascissa massima visualizzabile può arrivare da zero fino al valore 639 (=8 x 80). La modifica riguarda soprattutto la riga 250 in cui bisogna correggere il valore 160 ivi presente.
- Inserire la possibilità di cambiar colore (a patto di possedere la scheda grafica opportuna).

Ai possessori di C/128 (C/16 oppure Plus-4) consigliamo di:

- Inserire un'opzione per ottenere, contempora-

"Trasportare" su computer un problema di geometria o di matematica è uno dei compiti più ardui per un principiante

neamente, schermo grafico e di testo, in modo da non esser costretti a "swappare" (=scambiar tra loro) i due modi di visualizzazione per inserire nuovi dati.

PARABOLE

Come i più furbi avranno intuito, i programmi precedenti non erano altro che la prima fase per introdurre un argomento di maggiore interesse e, soprattutto, utile.

E' infatti piuttosto semplice tracciare segmenti avendo a disposizione le coordinate dei loro estremi: è certamente più facile scarabocchiare su un foglio di carta anzichè accendere il computer, caricare il programma, digitare Run ed inserire i dati.

Più complesso risulta, al contrario, tentare di disegnare a "mano libera" funzioni più sofisticate, tra cui le parabole.

Tutti sanno che la parabola è una particolare curva i cui punti soddisfano alla relazione matematica...

Y = a * x (exp) 2 + b * x + c

...in cui "a" e "b" sono i coefficienti dell'incognita e "c" è l'eventuale termine noto; "exp" significa: elevato a.

Nel caso in cui "a" sia nullo, l'equazione rappresenta non più una parabola, ma una retta.

Disegnare con carta e penna una qualsiai parabola, partendo dalla formula, è relativamente semplice. Supponiamo che questa sia...

Y = 3 * x (exp) 2 + 4 * x + 2

Di solito si considerano alcuni valori particolari di Y in corrispondenza di altrettanti valori particolari di X.

Quando X è nulla, ad esempio, Y vale 2; infatti...

Y = 0 + 0 + 2 = 2

Se, invece, X vale 1, Y vale...

Y = 3 + 4 + 2 = 9

Se X vale -1, Y vale...

Y = 3 - 4 + 2 = 1

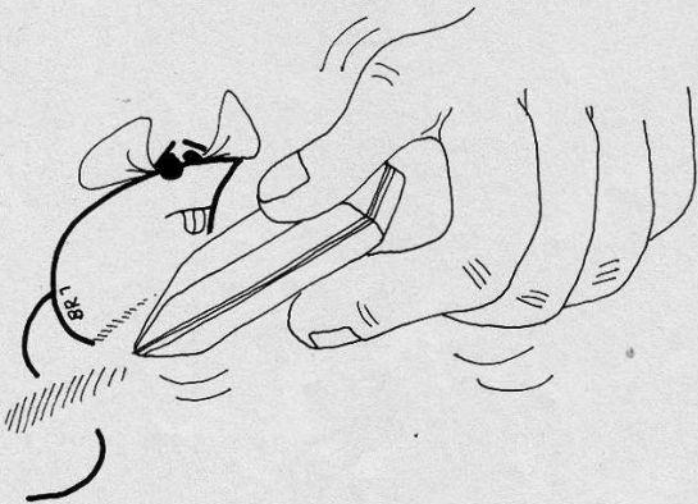
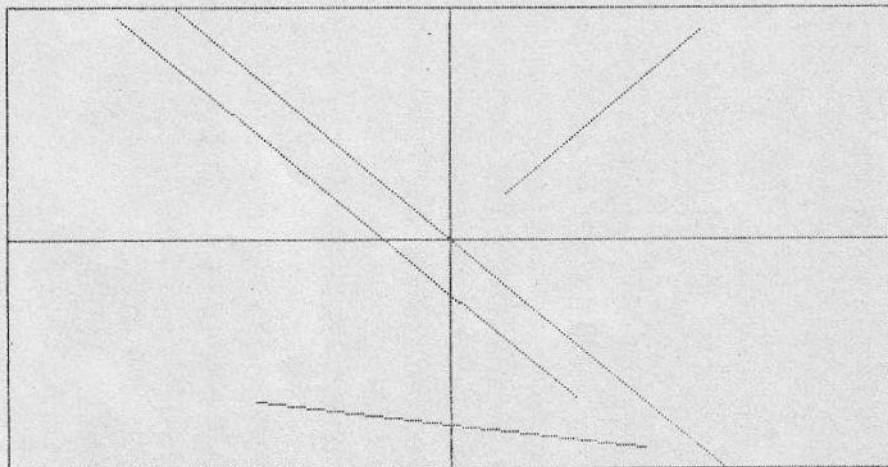
Con un po' di pazienza (e di tempo) è relativamente agevole tracciare sul foglio di carta un numero

sufficiente di punti che consentiranno, poi, di tracciare interamente la parabola cercata.

Il programma di queste pagine consente di risparmiar tempo e fatica dal momento che, introdotti i tre valori richiesti ("a", "b" e "c"), provvede a tracciare con notevole rapidità la curva cercata.

I listati pubblicati possono quindi servire per esaminare ciò che accade a mano a mano che aumenta il solo valore della costante; oppure a verificare il "peso" del coefficiente "b" rispetto ad "a"; oppure a vedere che succede se viene annullato "a" (e non "b") o viceversa; oppure...

Il computer, insomma, deve servire a qualcosa; nel caso specifico, il programma "Parabole" (riportato in queste pagine sia nella versione Ms-Dos sia in quella C/128 e C/16) può essere un validissimo strumento didattico, soprattutto considerando la sua estrema brevità e velocità di esecuzione.

Ecco, dunque, che emerge l'utilità del primo programma: se un problema presenta numerose difficoltà è bene affrontarle una per una e sfruttare l'esperienza acquisita in precedenza. "Segmenti" serviva solo a determinare, e sperimentare, l'elaborazione da introdurre per eliminare la diversità dei due sistemi di rappresentazione cartesiana (quello "classico", al quale siamo abituati, e quello del computer). Diventati padroni di tale tecnica, l'abbiamo inserita nel secondo programma.

Scrivendo direttamente un unico listato è facile cadere in errore e, spesso, non si riesce a capire se questo deve esser ricercato nell'errata soluzione del primo o del secondo problema.

```
100 REM SEGMENTI (VERSIONE C/128)
110 :
150 A=147:B=19
160 PRINT CHR$(A) :REM CANCELLA SCHERMO
170 PRINT CHR$(B);:INPUT"X1,Y1";X1,Y1
180 INPUT"X2,Y2";X2,Y2: REM COORDINATE
190 GRAPHIC 1,1: REM HI-RES
200 DRAW 1,0,0 TO 319,0 TO 319,199
210 DRAW TO 0,199 TO 0,0 :REM CORNICE
240 DRAW 1,0,100 TO 319,100:REM ASSE X
250 DRAW 1,160,0 TO 160,199:REM ASSE Y
260 REM CORREZIONE COORDINATE
270 DRAW 1,160+X1,100-Y1 TO 160+X2,100-Y2
280 GET A$:IF A$="" THEN 280
290 IF A$="." THEN GRAPHIC 0:END
300 GRAPHIC 0:GOTO 170
```

```
100 REM SEGMENTI IN GW-BASIC
110 :
120 PRINT"QUALE COMPUTER STAI USANDO?"
130 PRINT"1- C/64 + EMULATORE GW-BASIC"
140 PRINT"2- COMPUTER MS-DOS COMPATIB."
150 A=12:B=11:INPUT X:IF X=1 THEN A=147:B=19
160 PRINT CHR$(A):REM CANCELLA SCHERMO
170 PRINT CHR$(B);:INPUT"X1,Y1";X1,Y1
180 INPUT"X2,Y2";X2,Y2:REM COORDINATE
190 SCREEN 1:CLS 1:REM HI-RES
200 LINE(0,0)-(319,0),1:REM CORNICE
210 LINE-(319,199),1    :REM CORNICE
220 LINE-(0,199),1      :REM CORNICE
230 LINE-(0,0),1        :REM CORNICE
240 LINE(0,100)-(319,100),1:REM ASSE X
250 LINE(160,0)-(160,199),1:REM ASSE Y
260 REM CORREZIONE COORDINATE
270 LINE (160+X1,100-Y1)-(160+X2,100-Y2),1
280 A$=INKEY$:IF A$="" THEN 280
290 IF A$="." THEN END
300 SCREEN 0:GOTO 170
```

Per tracciare la parabola il programma provvede a disegnare numerosi segmenti legati l'un l'altro. La riga 290, comunque, si basa sui risultati dell'elaborazione di riga 270 (determinazione di un punto) e di riga 280 (determinazione del punto successivo al precedente, distante una lunghezza pari a "D").

Il programma (righe 180 - 210) chiede, oltre ai valori di "a", "b" e "c", anche il "passo" con cui si desidera tracciare la parabola; a passi elevati corrisponde una maggiore velocità di elaborazione ma una modesta precisione del grafico; con passi di modesta entità (anche minori di 1) gli effetti si invertono.

La richiesta di cancellare, o meno, lo schermo, consentirà di confrontare tra loro più curve disegnate in successione.

Suggerimenti

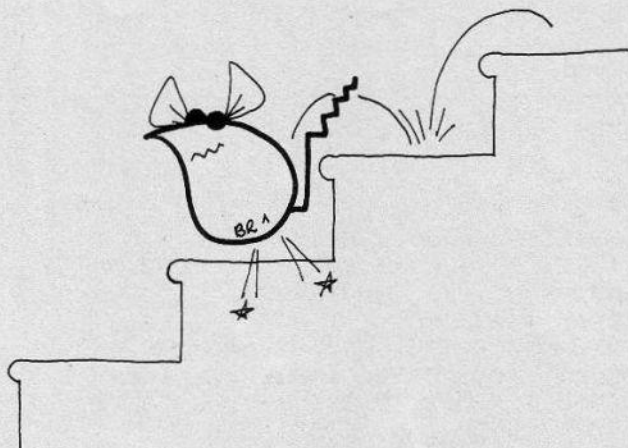
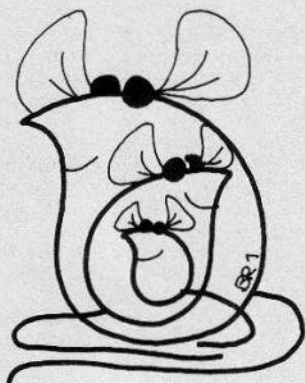
Oltre ai consigli già evidenziati per il programma "Segmenti", validi anche per "Parabole", provate a:

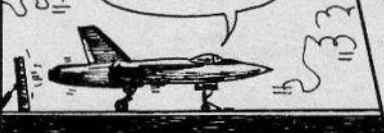
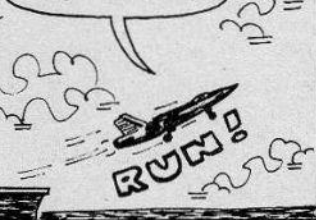
- Scrivere un programma che visualizzi una parabola il cui asse sia parallelo all'asse x (e non y, come in quelli pubblicati). La formula da prendere in considerazione non è più quella di riga 270 ma risulta leggermente diversa: al posto di x, infatti, bisogna inserire y, e viceversa.

- Sostituire la formula della parabola con quella del cerchio (o ellisse) o con altre funzioni, tra cui quelle trigonometriche; prevedere gli immancabili casi particolari (divisione per zero, punti all'infinito) e provvedere all'inserimento di opportuni "filtri" che impediscano l'interruzione dell'elaborazione.

CONCLUSIONI

Lavorare con un computer presuppone la disponibilità, da parte nostra, a risolvere un duplice problema: il primo è quello di individuare le relazioni matematiche (o le procedure logiche) necessarie per affrontare il problema stesso; il secondo, non meno importante, consiste nell'individuare la procedura più idonea per far digerire all'elaboratore i dati che verranno digitati.

Alla fine, però, la soddisfazione di veder funzionare correttamente un programma compensa ampiamente la fatica sopportata.

```
100 REM PARABOLE IN GW-BASIC
110 :
120 PRINT"QUALE COMPUTER STAI USANDO?"
130 PRINT"1- C/64 + EMULATORE GW-BASIC"
140 PRINT"2- COMPUTER MS-DOS COMPATIBILE"
150 Q=12:W=11:INPUT X:IF X=1 THEN Q=147:W=19
160 PRINT CHR$(Q):REM CANCELLA SCHERMO
170 PRINT CHR$(W):REM HOME
180 INPUT "COEFFICIENTE A";A
190 INPUT "COEFFICIENTE B";B
200 INPUT "COSTANTE     C";C
210 INPUT "STEP         D";D:IF D=0 THEN D=5
220 D$="S":INPUT "CLEAR SCREEN (S/N)";D$
230 SCREEN 1:IF D$="S" THEN CLS 1
240 LINE (0,100)-(319,100),1:REM ASSE X
250 LINE (160,0)-(160,199),1:REM ASSE Y
260 FOR X1=-160 TO 160-D  STEP D
270 Y1= A*X1*X1 + B*X1 + C:REM FORMULA
280 Y2= A*(X1+D)*(X1+D)+B*(X1+D)+C
290 LINE (160+X1,100-Y1)-(160+X1+D,100-Y2),1
300 NEXT
310 A$=INKEY$:IF A$="" THEN 310
320 IF A$="." THEN END
330 SCREEN 0:GOTO 170
```

Dopo aver spiegato a Petspeed che una memoria RAM perde tutte le sue informazioni ogni volta che viene spenta, Primo se ne torna al suo lavoro. Poco dopo, mentre il nostro eroe è al suo computer, un aereo è pronto sul ponte della portaerei...

MS-DOS, CHI ERA COSTUI?

Siete davvero sicuri che il mondo Ms-Dos sia misterioso e pieno di pericoli? Noi pensiamo che, al contrario, sia piuttosto simile al C/64; infatti...

di **Valentino Spataro**

Un PC è del tutto simile ad un C/64

L'italiano è storicamente e culturalmente uno dei popoli più attaccati alle tradizioni ed alle abitudini. Tale particolare si riflette positivamente a scuola (speriamo ancora per molto) nell'insegnamento del latino e dei Promessi Sposi, negativamente nella paura di fronte alle novità.

E' questa la situazione del sessantaquattrista che vede aumentare, ogni giorno di più, lo spazio dedicato ai PC compatibili ed all'Amiga. Il pericolo che si ritiene di correre, del tutto infondato, è dovuto principalmente al fatto che, dopo aver impiegato tante energie per capire un mondo nuovo, scopriamo tristemente che è gia "invecchiato" e crediamo di dover spendere altrettante energie per entrare in altro mondo nuovo.

Niente di più sbagliato.

In realtà, acceso il C/64 per la prima volta, tutti noi abbiano avuto due grandi problemi: capire ciò che avevamo davanti e... usarlo. Entrando nel mondo dei PC, invece, non resta che il secondo problema.

Scopo del presente articolo è, quindi, entrare nel nuovo mondo da esperti(!) quali siamo, lasciando da parte qualsiasi timore reverenziale. La lettura è consigliata anche a chi non sa niente di Amiga-dos perchè questo sistema operativo affonda le sue radici nell'MS-DOS.

ACCENDIAMO UN PERSONAL

Se è vero che un C/64, da solo, può far poco, un personal sa fare ancora meno; è dotato, però, di maggiori potenzialità.

Quando accendiamo un C/64 abbiamo a disposizione subito (perchè residenti su ROM) un interprete Basic, un dispositivo per la gestione delle periferiche ed altre cose di cui, in questa sede, non si ha interesse a parlare.

Quando, invece, accendiamo un PC (privo di disco rigido e senza introdurre un particolare dischetto) il primo messaggio che appare è un messaggio di errore, perchè il computer, così com'è, non è in grado di fare assolutamente niente: mancano, infatti, le routine per la gestione delle periferiche ed un qualunque linguaggio, come il Basic. A questa mancanza supplisce il dischetto del DOS, da inserire prima di accendere l'apparecchio, e che viene caricato automaticamente al momento dell'accensione.

Il DOS (Disk Operative System: Sistema Operativo su Disco) è l'insieme di routine necessarie per gestire correttamente le risorse del computer. Le uniche cose che possiamo fare sono: impartire comandi "diretti" nel linguaggio del DOS, oppure caricare, ed eseguire, programmi scritti in DOS o in linguaggio macchina.

Evidentemente, quando accendiamo un C/64, non abbiamo di questi problemi; non abbiamo, però, neanche la possibilità di cambiare il Kernal oppure i dispositivi di Input / Output così velocemente come si può fare su un PC, vale a dire inserendo, all'accensione, un disco piuttosto che un altro. Da queste caratteristiche deriva il successo delle varie versioni di DOS che hanno permesso a questi apparecchi di non invecchiare alla stessa velocità degli altri computer.

Procediamo: acceso un C/64 compare la scritta "READY." ed il cursore lampeggiante; caricato il DOS su un PC, invece, dopo le richieste di data odierna e orario attuale, compare una (scarsamente) indicativa lettera dell'alfabeto seguita dal segno maggiore(>), per esempio A> oppure C>, denominate, in gergo, "prompt".

In entrambi i casi i computer, visualizzato il prompt, sono in attesa di ricevere comandi da parte dell'utente.

Sul C/64 possiamo caricare un programma

(LOAD "NOME PROGRAMMA", 8), effettuare operazioni matematiche (PRINT 3*5) o eseguire comandi diretti (LIST): possiamo cioè lavorare con il computer e con le periferiche.

Da DOS, invece, possiamo caricare programmi o eseguire comandi del DOS: a differenza del C/64, quindi, non possiamo lavorare con il computer in modo diretto (per esempio fargli fare calcoli senza usare programmi specifici).

COME UTILIZZARE LE PERIFERICHE

Il prompt del DOS indica con quale periferica stiamo lavorando in quel particolare momento.

Chiariamo questo concetto facendo alcuni esempi di utilizzo delle periferiche su C/64. Supponiamo di avere in memoria un programma Basic e di volerlo stampare; dovremo scrivere, in modo diretto...

OPEN 2, 4: CMD 2: LIST

Dobbiamo cioè "aprire" il dispositivo interessato (4 = stampante) e svolgere le operazioni che servono (CMD 2: LIST). Operando con il Dos, invece, basta selezionare il dispositivo su cui intendiamo lavorare: il computer provvederà, da solo, ad aprirlo e chiuderlo di volta in volta, ad ogni comando che impartiamo.

Nel C/64 i numeri dei dispositivi sono già definiti all'accensione: 1 per il registratore; 4 e 5 per la stampante; 8, 9, 10 e 11 per i 4 differenti drive; e così via. Nei PC è (quasi) la stessa cosa: la lettera del prompt indica su quale dispositivo stiamo operando:

"A" e "B" sui drive a floppy disk; "C", "D" e "E" sui dischi rigidi (oppure Ram disk); LPT1 per la stampante. Per cambiare dispositivo basterà scrivere, subito dopo il prompt, la lettera che individua il nuovo dispositivo, seguita dal carattere di doppio punto (Esempio, B:) e premere il tasto Return; il prompt diventerà: B>. Le sigle (A, B, C, LPT1, etc.) possono però essere anche cambiate a piacere dall'utente, fatto che conferma la maggiore flessibilità del DOS rispetto al C/64. Da ricordare, ancora, che il prompt, all'accensione, corrisponde al dispositivo da cui è stato caricato il DOS.

LE PERIFERICHE, CARATTERISTICHE

Le periferiche dei PC sono le stesse di un C/64 ma, ovviamente, molto più veloci e sofisticate. Possono essere drive, dischi rigidi, stampanti, modem, e così via; ci riferiremo, ora, solo ai drive ed ai dischi rigidi (hard disk).

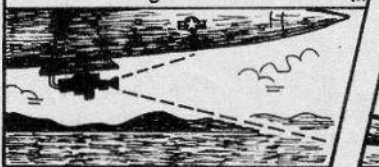
Differenze a livello concettuale, tra drive e dischi rigidi, non ve ne sono: entrambi usano dischi magnetici su cui memorizzano le informazioni. Le differenze stanno nella capacità di contenere dati e nella velocità di lettura e scrittura, caratteristiche tutte a favore degli hard disk. Gli svantaggi, invece, sono da individuare nel maggior prezzo di acquisto (ma, date le prestazioni, sono più che convenienti) e nell'impossibilità di trasportarli, essendo fissi (vengono infatti chiamati anche dischi fissi).

Per ovviare ai due problemi esistono dischi rigidi portatili, ma da utilizzare sui PC appositamente predisposti; altra soluzione è quella di registrare i dati contenuti nel disco rigido su cassette particolari (i cui lettori sono chiamati "streamer"), trasportare la cassetta e registrarli su altro disco rigido.

In entrambi i casi il problema maggiore è la mancanza di uno standard comune a tutti gli apparecchi. E' bene aprire una parentesi in proposito: uno degli argomenti a favore del C/64 è, ed è stato, la grande quantità di programmi in circolazione. Entrando nel mondo dei PC, bisogna fare attenzione non solo ad avere una macchina 100% compatibile MS-DOS, ma anche alle periferiche che si acquistano. Se non appartengono a standard di mercato collaudati, si rischia di restare con macchine dalle prestazioni magari eccellenti, ma che al primo guasto possono portare alla paralisi del nostro lavoro se l'assistenza è scarsa.

Sempre a proposito della compatibilità bisogna fare attenzione anche al formato dei floppy; l'evoluzione della tecnica ha permesso maggiori diponibilità di dati in superfici sempre più piccole e più protette (quali i dischi da 3.5 pollici). Da questo particolare nasce il problema delle capacità dei floppy e della loro formattazione.

Uno stesso dischetto, infatti, può essere formattato in modi diversi, a seconda delle esigenze; può quindi capitare di non poter leggere il floppy che abbiamo tra le mani. Il motivo del mancato funzionamento è uno dei seguenti:
1) il formato: lettori da 5.25 non leggono dischi da

Un C/64, appena acceso, può solo svolgere qualche operazione; un PC nemmeno quello!

Un C/64, appena acceso, può solo svolgere qualche operazione; un PC nemmeno quello!

3.5 e viceversa (ovvio limite fisico!)
2) un drive può non essere capace di leggere dischetti il cui contenuto sia superiore ad una certa capacità a causa di limiti meccanici (limite hardware)
3) un determinato drive, pur essendo in grado, tecnicamente, di leggere anche i dischi più capaci, può non riconoscere determinate formattazioni perchè la versione DOS usata non è in grado di riconoscerle (limite software).

Vale, fortunatamente, il principio secondo cui versioni più recenti del DOS sono totalmente compatibili con quelle precedenti nella lettura (ma, a volte, non nella scrittura).

Per ciò che riguarda i dischi rigidi c'è da dire che il DOS non riesce a gestire più di 30 Mb (megabyte) per volta. Nel caso il disco rigido abbia una capacità superiore (ad esempio, 70 Mb) il DOS, formattando il disco rigido (operazione necessaria come nei floppy) lo divide automaticamente in tre parti contraddistinte, ciascuna, da diverse lettere dell'alfabeto: la C, i primi 30 Mb; la D, i successivi 30; la E i restanti 10 Mb. Pare, comunque, che la prossima versione del DOS, la V 4.0, sarà in grado di formattare con una sola lettera dischi rigidi almeno fino ad 1 gigabyte (circa un miliardo di byte!).

E' doverosa un'ultima precisazione a proposito di come il DOS riconosce i drive, caso che avviene in maniera diversa rispetto al popolare 1541. Abbiamo già considerato come un unico disco rigido sia visto dal DOS come una pluralità di periferiche (un disco da 70 Mb equivale a 3 periferiche); al contrario, uno stesso drive può essere visto come due drive: questo capita generalmente con programmi complessi che, necessitando di più di floppy, ricorrono forzatamente alla simulazione accennata.

ALCUNI COMANDI COMUNI

Desiderio di tutti è poter mettere subito le mani sul computer e utilizzarlo senza perdere tempo: ecco quindi alcuni comandi essenziali, ben conosciuti ai possessori del 1541, ma ancora misteriosi in DOS. Ad esempio, per chiedere la directory di un disco basta selezionare la periferica che interessa e scrivere DIR (seguito dall'immancabile tasto return).

Così facendo viene fornito il nome dei file memorizzati, la loro lunghezza, il tipo di file, la data e l'ora di registrazione.

Se, invece, interessa soltanto esaminare i nomi dei file presenti su supporto magnetico, basterà scrivere DIR/W. Per avere, ancora, tutte le informazioni visualizzate con DIR, una pagina alla volta, si dovrà scrivere DIR/P. E' possibile inoltre vedere tutti i file il cui nome abbia determinati elementi comuni, come nel 1541: per esempio i comandi...

LOAD "$:P?IPPO*", 8
LIST

...del C/64, equivalgono al comando DOS:
DIR P?PPO*

Per formattare un disco si scrive...
FORMAT A:
...in cui, al posto di A, si può mettere B, C, D, come già visto. La procedura è necessaria per qualsiasi disco si voglia usare per la prima volta, sia esso un floppy o un disco rigido. Equivale al comando...
OPEN 15, 8, 15, "N:NOME DISCO,ID"
...del C/64 collegato ad un comune 1541.

Per copiare un programma è disponibile un'istruzione dedicata (simile a quella del 1541), COPY.

Supponendo che il prompt sia A>, si scriverà...
A> COPY nomefile B:

Così facendo il programma "nomefile" viene copiato da un'unità ad un'altra (nel nostro caso da A a B). Nel caso si disponga di un'unica unità, si dovrà sostituire il disco origine con quello destinazione (come avviene, insomma, con tutti i copiatori per il 1541). Infine, per caricare un programma è sufficiente digitare il suo nome e premere il tasto return.

Un ultimo esempio di versatilità conclude queste note: se, per esempio, state lavorando con un programma di videoscrittura che gira sul disco rigido, i testi digitati vanno, di solito, automaticamente salvati sullo stesso disco rigido. Per salvarli su un altro drive basta premettere, al nome del file da registrare, la lettera della periferica corrispondente seguita dal carattere di doppio punto; per esempio con...
A: PIPPO
...il testo chiamato PIPPO verrà caricato (o salvato), a seconda dell'operazione richiesta, sul minidisco contrassegnato dalla lettera A.

Con un C/64, invece, se un programma non prevede esplicitamente la possibilità di utilizzare due drive, la facilitazione appena esaminata non può essere sfruttata.

Potremmo continuare con tanti altri esempi che, inevitabilmente, risulterebbero complessi.

Il nostro scopo era solo quello di far capire, soprattutto a chi dispone del solo C/64, che un PC non è altro che... un computer, del tutto simile al calcolatore che già possediamo, ed usiamo, da tempo.

FACCIA A FACCIA CON L'HI-RES

Una succulenta manciata di Poke e Sys per accedere direttamente, anche in linguaggio macchina, alle potenti routine grafiche del C/128

di **Domenico Pavone**

I comandi grafici del C/128 sono accessibili direttamente da linguaggio macchina

Uno dei motivi d'orgoglio dei piccoli Commodore, è sempre stata la possibilità di utilizzare una grafica in alta risoluzione.

L'avvento del C/64, a suo tempo, ha praticamente rivoluzionato il mercato, nonostante la discreta difficoltà (da parte dell'utente) di gestire tale risorsa.

Com'è noto, infatti, nel suddetto computer non è implementato alcun comando Basic dedicato all'hi-res, esclusivo dominio del linguaggio macchina o di utility ad essa riservate (impossibile non citare le ancora più che valide Routine di Toma ed il successivo emulatore Gw-Basic).

Con il C/128, la situazione si è praticamente invertita.

L'introduzione di comandi come Graphic, Circle, ecc., ha reso talmente facile e veloce il compito del programmatore, da far sembrare assurdo il ricorso alla montagna di Poke necessarie per addomesticare l'alta risoluzione.

Tuttavia... non si vive di solo Basic (o almeno, non tutti intendono viverci), e, tra l'altro, una via di mezzo esiste: operare in linguaggio macchina, sfruttando le routine già presenti nell'interprete Basic del C/128.

Vediamo dunque come e dove accedere a queste routine, specificamente riferendoci al modo bitmap standard (il multicolor verrà solo accennato nelle sue eventuali differenze).

Come esempio, si considerino anche i brevissimi listati Basic pubblicati in queste pagine, ma soprattutto il disassemblato commentato, sfruttabile anche tramite il relativo caricatore Basic (quello con Read...Data).

Una sola avvertenza: qualora si volesse utilizzare il monitor per copiare il disassemblato (sempre meglio che digitare la noiosa sequela di numeri), si ricordi che è necessario porre in bank 1 (da $2000 in poi) la stringa relativa al comando Char.

COLOR

Per l'assegnazione dei colori di sfondo e primo piano non è necessario accedere alla routine del Basic vera e propria, per la cronaca allocata a partire dalla locazione esadecimale $69E2, 27106 in decimale (di banco 15, come tutti gli indirizzi riportati da ora in avanti).

E' sufficiente, infatti, settare opportunamente la locazione $302 (decimale 994) in modo che il suo nibble alto (= i quattro bit più significativi) contenga il valore da 0 a 15 indicante il colore di primo piano, mentre nel nibble basso andrà inserito il colore di sfondo.

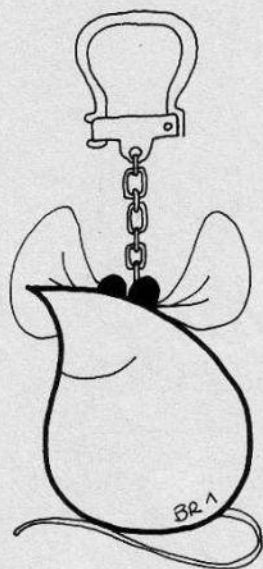
Nelle prime due istruzioni del disassemblato, per esempio, viene depositato in questa locazione il valore 15 ($0F), che in binario corrisponde a 0000 1111.

Si avrà quindi il colore 0 (nibble alto) per il disegno, ed il colore 15 per lo sfondo, corrispondenti a nero e grigio chiaro.

Si tenga presente che il numero del colore non corrisponde a quello riportato sul manuale a proposito dell'istruzione Color, ma a quello che normalmente si adopera per le note locazioni 53280 o 53281 (in pratica, lo stesso di Color, diminuito di una unità).

Quanto detto, è valido solo per il modo bitmap standard, mentre per il multicolor è necessario ricorrere alla locazione $303 (decimale 995), inserendo nel nibble alto il colore sorgente 1 ed in quello basso il colore sorgente 2 (per lo sfondo si può sempre ricorrere alla 53280).

Organizzata opportunamente la locazione $302 (o $303), è poi indispensabile assegnare il colore sorgente, quello che in Basic costituisce il primo parametro presente dopo Color.

Come si può notare dalle istruzioni 1300 - 1307

```
DISASSEMBLATO COMMENTATO
1300 LDA #$0F    Attributi sfondo
1302 STA $03E2   e primo piano.
1305 LDX #$01    Assegna colore per
1307 STX $83     grafica in uso.
-------------------------------------
1309 JSR $9F4F   Allocazione bitmap.
130C JSR $6A9B   Scnclr 1.
-------------------------------------
130F LDA #$20    Attivazione bitmap
1311 STA $D8     (graphic 1).
-------------------------------------
1313 LDA #$A0    Coordinata X
1315 STA $1150   (orizzontale)
1318 LDX #$00    centro del
131A STX $1151   cerchio.
131D LDA #$64    Coordinata Y
131F STA $1152   (verticale) centro
1322 STX $1153   del cerchio.
1325 LDA #$14    Definizione
1327 STA $1154   del raggio.
132A JSR $66B1   Ingresso a Circle.
-------------------------------------
132D LDA #$50    Coordinata X pixel
132F STA $1150   superiore sinistro.
1332 LDA #$1E    Coordinata Y pixel
1334 STA $1152   superiore sinistro.
1337 LDA #$C8    Coordinata X pixel
1339 STA $115C   inferiore destro.
133C LDA #$96    Coordinata Y pixel
133E STA $115E   inferiore destro.
1341 LDX #$00    Azzeramento
```

Mentre Jack Assembler, irritato dal pessimo funzionamento della scheda grafica montatagli da Giovedini, fa ritorno alla portaerei, il buon Primo è ancora ignaro di tutto...
Egli si trova nella sua cabina e sta discutendo...

```
1343 STX $1151   dei byte alti
1346 STX $1153   per le coordinate
1349 STX $115D   superiore sinistra
134C STX $115F   ed inferiore destra.
134F STX $1154   Angolo di rotazione.
1352 JSR $62D7   Ingresso a Box.
--------------------------------------
1355 LDA #$00    Posizionamento
1357 STA $1131   orizzontale.
135A LDA #$64    Posizionamento
135C STA $1133   verticale.
135F JSR $61D0   Ingresso a Fill.
--------------------------------------
1362 LDA #$3C    Coordinata X per
1364 STA $1131   pixel dipartenza.
1367 LDA #$64    Coordinata Y per
1369 STA $1133   pixel di partenza.
136C STA $1137   Coord. Y di arrivo.
136F LDA #$FF    Coordinata X per
1371 STA $1135   pixel di arrivo.
1374 JSR $9B30   Ingresso a Draw Line.
--------------------------------------
1377 LDA #$0D    Posiziona colonna
1379 STA $115E   per stringa.
137C LDX #$16    Posiziona riga
137E STX $115F   per stringa.
1381 LDA #$0E    Len stringa.
1383 LDX #$00    Indirizzo (low/hi)
1385 LDY #$20    stringa (in bank 1).
1387 JSR $6815   Ingresso a Char.
--------------------------------------
138A RTS         Return.
```

AIUTARE I BISOGNOSI

Volendo definire lo sparuto (ma non troppo) drappello dei frustrati utenti del Commodore 128, non si può fare a meno di immaginare i proverbiali prigionieri di una gabbia dorata.

Pur disponendo, infatti, della migliore macchina tra gli "otto bit" in commercio, quella dalle prestazioni potenzialmente più interessanti, inevitabilmente si è costretti a scontrarsi con la cronica, e forse definitiva, scarsità di software.

- Poco male - si saranno detti i non pochi reduci da fatiche sessantaquattriste. - Per il software c'è sempre disponibile il C/64 incorporato, al modo 128 ci penso io! -

Ed eccoci al secondo, e forse più cocente, motivo di frustrazione: sulle varie riviste, ed i nostri lettori non hanno certo bisogno di cercare troppo lontano, qualcosa effettivamente si trova.

Ma in quanto a documentazione ufficiale un po' più approfondita, neanche a parlarne.

Esiste, è vero, una quasi introvabile Programmer's Reference Guide, ma rigorosamente in inglese, e comunque dai contenuti non troppo "sfiziosi" per uno smanettone incallito.

Non rimane, dunque, che affidarsi al monitor LM di cui è equipaggiato il nostro C/128 per "rovistare" tra le ROM del sistema.

Se, tuttavia, la pigrizia o la non perfetta conoscenza del linguaggio macchina vi impediscono di farlo, niente paura: ecco per voi una prima serie (ne seguiranno altre) di indirizzi e locazioni utilizzabili prevalentemente in linguaggio macchina, ma, soprattutto, che possono servire come punto di partenza per ricerche più approfondite.

Basta qualche Poke ed una Sys per ottenere gli stessi effetti del Basic

del disassemblato, la cosa risulta alquanto semplice: basta inserirlo nella locazione $83 (decimale 131).

Per la normale grafica a matrice di punti "pokeremo" dunque 0, oppure 1, per scegliere se disegnare una figura con il colore di sfondo o con quello di primo piano, mentre in multicolor potrà anche essere usato un valore 2 o 3.

GRAPHIC E SCNCLR

Con il comando Graphic, in Basic, viene selezionato ed attivato il tipo di grafica sul quale agire, con la possibilità di cancellare lo schermo attivo.

La routine che viene eseguita allorchè lo si impartisce, in effetti, compie anche un altro importantissimo lavoro: l'allocazione dell'area di memoria riservata all'alta risoluzione.

In pratica, com'è noto, l'inizio della RAM utilizzabile dai programmi Basic (in bank 0) viene spostato da $1C00 (decimale 7168) a $4000 (16384), trasferendo nella nuova area anche l'eventuale programma residente in memoria.

In linguaggio macchina, dovremo assolvere noi a questo compito con un semplice salto (tramite JSR) all'indirizzo $9F4F (decimale 40783).

L'operazione inversa, corrispondente a Graphic Clr, viene compiuta dalla routine di sistema posta in $A022 (decimale 40994).

Per eseguire anche la pulizia della pagina grafica, si può sfruttare la routine preposta all'esecuzione di Scnclr, o meglio, un suo possibile ingresso: $6A9B (decimale 27291), da richiamare con il registro X contenente 1 per il modo bitmap, 3 per il multicolor, ecc. (come il parametro Basic, insomma).

Seguendo ancora il disassemblato, eccoci ora alle righe con indirizzo 130F - 1311, che provocano il passaggio in Hi-Res modo bitmap standard (Graphic 1, per intenderci).

In effetti, il comando Basic Graphic ha una sua routine con ingresso a $6B5A (decimale 27482), ma risulta molto più pratico (sempre se non si vuole ricorrere ai normali registri del VIC) utilizzare la locazione $D8 (decimale 216), o meglio i suoi tre bit più alti, inserendo i seguenti valori:

0 ($00) = Graphic 0
32 ($20) = Graphic 1
96 ($60) = Graphic 2

160 ($A0) = Graphic 3
224 ($E0) = Graphic 4

Volendo usare lo schermo diviso (split screen) di Graphic 2 (o Graphic 4), la riga di demarcazione può essere stabilita manipolando la locazione $A34 (decimale 2612), il cui contenuto viene copiato (solo dopo aver settato il tipo di grafica!) nel Raster Register per determinare da quale linea di scansione raster iniziare la visualizzazione dello schermo di testo (un articolo sulla gestione del Raster del C/128 è apparso sul numero scorso).

La linea di scansione raster, e quindi il valore da depositare nella locazione $A34, si ottiene moltiplicando per 8 la riga di schermo e poi sommando 48 al valore ottenuto.

```
10 REM -------------------
20 REM       CIRCLE
30 REM -------------------
40 BANK15:X=4432:POKE994,15
50 GRAPHIC1,1:POKE131,1
60 POKEX,160:POKEX+1,0
70 POKEX+2,100:POKEX+3,0
80 POKEX+4,90:SYS26289
90 GETKEYA$:GRAPHIC0:END
```

CIRCLE

Per eseguire Circle il sistema utilizza una routine posta a partire da $668E (decimale 26254), la cui prima parte è però dedicata all'accettazione ed interpretazione dei vari parametri (dal testo Basic).

Per i nostri scopi (1313 - 132A del disassemblato), un ingresso possibile è $66B1 (decimale 26289), usando per i parametri le seguenti coppie di locazioni (formato basso / alto):
In $1150 - $1151 (decimale 4432 - 4433) la coordinata orizzontale del centro del cerchio, espressa in pixel.

```
100 REM ---------------------------------------------------
110 REM        HIRES IN LINGUAGGIO MACCHINA SUL C/128
120 REM ---------------------------------------------------
130 FAST:BANK1:FORX=0TO13:READA$:POKE8192+X,ASC(A$):NEXT
140 BANK15:FORX=0TO138:READA:POKE4864+X,A:CK=CK+A:NEXT
150 SLOW:IFCK<>13421THENPRINT"ERRORE NEI DATA!":END
160 SYS4864:GETKEYA$:GRAPHIC0:END
170 :
180 DATA "P","R","E","M","I"," ","U","N"," ","T","A","S","T","O"
190 DATA 169,015,141,226,003,162,001,134,131,032,079,159,032,155
200 DATA 106,169,032,133,216,169,160,141,080,017,162,000,142,081
210 DATA 017,169,100,141,082,017,142,083,017,169,020,141,084,017
220 DATA 032,177,102,169,080,141,080,017,169,030,141,082,017,169
230 DATA 200,141,092,017,169,150,141,094,017,162,000,142,081,017
240 DATA 142,083,017,142,093,017,142,095,017,142,084,017,032,215
250 DATA 098,169,160,141,049,017,169,100,141,051,017,032,208,097
260 DATA 169,060,141,049,017,169,100,141,051,017,141,055,017,169
270 DATA 255,141,053,017,032,048,155,169,013,141,094,017,162,022
280 DATA 142,095,017,169,014,162,000,160,032,032,021,104,096
290 END
```

```
10 REM -------------------
15 REM          BOX
20 REM -------------------
25 BANK15:X=4432:POKE994,15
30 AN=0:REM -->ANGOLAZIONE
35 FL=0:REM 0=VUOTO 1=PIENO
40 GRAPHIC1,1:POKE131,1
45 POKEX,80:POKEX+1,0
50 POKEX+2,30:POKEX+3,0
55 POKEX+12,200:POKEX+13,0
60 POKEX+14,150:POKEX+15,0
65 POKEX+4,AN:POKEX+5,0
70 SYS25303,,FL
75 GETKEYA$:GRAPHIC0:END
```

```
10 REM -------------------
20 REM   DRAW (LINE)
30 REM -------------------
40 BANK15:X=4401:POKE994,15
50 GRAPHIC1,1:POKE131,1
60 POKEX,60:POKEX+1,0
65 POKEX+2,100:POKEX+3,0
70 POKEX+4,255:POKEX+5,0
75 POKEX+6,100:POKEX+7,0
80 SYS39728
90 GETKEYA$:GRAPHIC0:END
```

```
10 REM -------------------
20 REM      DRAW (POINT)
30 REM -------------------
40 BANK15:X=4401:POKE994,15
50 GRAPHIC1,1:POKE131,1
60 POKEX,160:POKEX+2,100
80 SYS39931
90 GETKEYA$:GRAPHIC0:END
```

In $1152 - $1153 la coordinata verticale del centro.
In $1154 - $1155 la lunghezza del raggio (sempre in pixel).

BOX

Ingresso "ufficiale" a $62B7 (decimale 25271), ma realmente utilizzabile (nel disassemblato = 132D - 1352) a partire dall'indirizzo $62D7 (decimale 25303).

I parametri vanno depositati nelle stesse locazioni usate da Circle, e precisamente:
$1150 - $1151 = Coordinata orizzontale dell'angolo superiore sinistro.
$1152 - $1153 = Coordinata verticale dell'angolo superiore sinistro.
$1154 - $1155 = Angolo di rotazione della figura.
$115C - $115D = Coordinata orizzontale dell'angolo inferiore destro.
$115E - $115F = Coordinata verticale dell'angolo inferiore destro.

Prima del salto all'indirizzo citato, è necessario anche preparare il registro X. Caricato con 0, verrà tracciato solo il perimetro del poligono, con 1, invece, lo stesso risulterà "pieno".

PAINT

Per il riempimento di aree delimitate dello schermo hi - res, è sufficiente inserire le coordinate del punto di partenza nelle locazioni $1131 - $1132 (decimale 4401 - 4402) e $1133 - $1134, nell'ordine quella orizzontale e quella verticale (sempre in formato low/hi e con valore inteso come numero di pixel), quindi saltare all'indirizzo $61D0 (decimale 25040).

DRAW

Il Basic 7.0 usa come ingresso per questa routine $6797 (decimale 26519), dove ha inizio la solita valutazione dei parametri che seguono il comando.

Per accedervi da linguaggio macchina, gli indirizzi utili sono due, a seconda che occorra plottare un singolo punto o tracciare una linea.

Iceman apprezza molto la mascotte di Primo e si dimentica le sue strane idee in fatto di "emulatori"...
Intanto sono passati quattro o cinque cicli di clock e Jack Assembler ha fatto ritorno sulla portaerei.
Giovedini gli si reca incontro...

Nel primo caso occorre inserire le coordinate del punto nelle stesse locazioni già viste per Paint (da $1131 a $1134), quindi saltare all'indirizzo $9BFB (decimale 39931).

Per il tracciamento, invece, nelle locazioni da $1131 a $1134 vanno depositate le coordinate orizzontale e verticale del punto di inizio della retta, mentre per il punto di arrivo si utilizzano le quattro successive: in $1135 - $1136 la coordinata orizzontale, in $1137 - $1138 quella verticale.

Per il salto alla routine, infine, si può utilizzare l'ingresso $9B30 (decimale 39728), come nel caso del nostro disassemblato (istruzioni 1362 - 1374).

CHAR

Anche l'uso di caratteri in ambiente hi - res risulta estremamente semplice, tenendo però presente che in questo caso le coordinate si intendono non più in pixel ma in numero di colonna (quella orizzontale, compresa tra 0 e 39) e numero di riga (quella verticale, compresa tra 0 e 24).

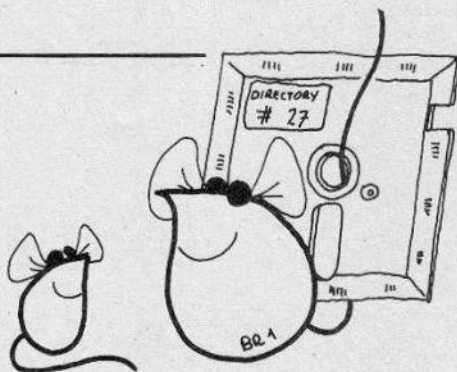
Colonna e riga vanno rispettivamente "storate" nelle locazioni $115E (decimale 4446) e $115F (decimale 4447), per indicare il punto dal quale inizierà la visualizzazione della stringa.

Quest'ultima dovrà essere presente (= la dovremo mettere noi) in un qualsiasi indirizzo di bank 1.

Il salto alla routine sarà diretto verso $6815 (decimale 26645), con i registri del microprocessore così preparati:
in accumulatore il numero di caratteri della stringa;
in X il byte basso dell'indirizzo di bank 1 in cui è presente la stringa;
in Y il byte alto dello stesso indirizzo.

Nel disassemblato, alle righe di istruzioni 1377 - 1387, si sono utilizzate per la stringa le locazioni di bank 1 da $2000 in poi (la stringa è letta e pokata tramite i read / data del listato Basic), ma con una certa... disinvoltura.

Nel senso che non si è prevista alcuna interferenza delle variabili del Basic in quelle locazioni.

Nella pratica, è più prudente "riservarsi" una piccola fetta dell'immenso bank 1 alterando, per esempio, i puntatori di inizio dell'area utilizzata dalle variabili ($2F - $30).

```
10 REM -------------------
15 REM          CHAR
20 REM -------------------
25 BANK1:FORX=0TO13:READA$
30 POKE8192+X,ASC(A$):NEXT
35 BANK15:POKE994,15
40 GRAPHIC1,1:POKE131,1
45 POKE4446,13:POKE4447,22
50 SYS26645,14,0,32
55 GETKEYA$:GRAPHIC0:END
60 DATA "P","R","E","M","I"
65 DATA " ","U","N"," ","T"
70 DATA "A","S","T","O"
```

ULTIME NOTE

Praticamente è tutto, con una sola eccezione: il comando Locate.

Ebbene, non è stato dimenticato, ma risulta del tutto... inutile.

Già in Basic la sua funzione è abbastanza limitata, ma sfruttabile soprattutto grazie alla sua interazione con RDOT(n) o alla possibilità, usando i vari comandi, di ometterne alcuni parametri.

In linguaggio macchina, come appena visto, vanno obbligatoriamente indicate le coordinate per tutte le routine, e comunque RDOT non ha molto valore.

Se, poi, volete tentare strade di maggiore difficoltà, l'ingresso per Locate (quello di sistema) è posto a $6855 (decimale 26965), e sfrutta le già citate locazioni da $1131 a $1134.

A voi il compito di approfondire l'argomento (e magari segnalarcelo), se pensate che ne valga la pena.

Chi lavora in l.m. può, finalmente, svolgere lavori completi senza chiedere aiuto al Basic

UN ANTI-VIRUS PER IL TUO DRIVE

Presentiamo una tecnica di auto-difesa per annullare gli effetti del sabotaggio provocato da un terribile virus, molto diffuso, che colpice i drive 1541 in modo apparentemente irreparabile

di **Franco Rodella**

Il tallone d'Achille del 1541 è rappresentato da un mancato controllo da parte del suo s.o.

Prima di parlare di come è possibile "bloccare" un drive, è necessario capire come funziona il sistema operativo del 1541.

Quanto stiamo per dire è valido per i modelli 1541, 1570 e 1571, utilizzato, quest'ultimo, in "modo" 1541.

La superficie fisica di un dischetto formattato è divisa in varie piste, chiamate tracce. Il drive può arrivare a selezionarne fino a 41, ma il DOS ne utilizza, solitamente, solo 35.

Sfruttando tale limitazione possiamo ingannare il disk drive, costringendolo a posizionare la testina in una delle zone non usate (come la traccia n. 41) in modo che la periferica non sia più in grado di riconoscere i dischetti inseriti.

Per rendere efficace il virus, gli "untori" hanno escogitato un sistema che sfrutta la particolarità per la quale il drive "dimentica" di aver portato la testina al di fuori dalla zona standard. Si potrebbe obiettare che l'utente, costretto a spegnere e riaccendere la periferica (oppure, per i fortunati possessori del C/128-D, resettarla) possa risolvere il problema.

In effetti, ad un primo esame, si è indotti a pensare che il DOS, dopo un reset, riporti semplicemente la testina nella zona "normale" (tracce 1 : 35) prima di effettuare una qualunque altra operazione.

Il drive, invece, non riposiziona la testina nella zona standard perchè, al momento dell'accensione, il 1541 non pensa assolutamente che la testina possa essere posizionata oltre la traccia 35; ne consegue che il sistema operativo non si preoccupa minimamente di farla rientrare!

POSIZIONANDO

Vediamo ora come è possibile fare in modo che la testina si posizioni sulla traccia 41, visto che normalmente questa traccia non è accessibile con operazioni normali.

Per tale scopo useremo due locazioni della memoria del drive: la locazione 0 e la locazione 6.

Queste sono usate dal 1541 proprio per spostare la testina, per leggere, per scrivere e per svolgere, insomma, molte operazioni sul dischetto.

Dal momento che solitamente è il DOS che accede direttamente alle suddette locazioni, non viene fatto alcun controllo sulla validità dei dati presenti, e quindi è possibile scrivervi qualunque valore. Ne possiamo quindi approfittare per spostare la testina nella "zona proibita", senza alcun problema.

Finora abbiamo parlato di modifiche nella memoria del drive, ma forse non tutti sanno come effettuarle.

E' possibile scrivere dati nella memoria interna del 1541 usando il comando M-W, che equivale al comando Basic POKE; anzi, risulta addirittura più potente, visto che è possibile scrivere fino a 34 valori con un solo comando!

Affinchè, ad esempio...

Poke 0, 128

...agisca sulla memoria del drive, il comando deve essere scritto nella forma seguente:

"m - w" + chr$(0) + chr$(0) + chr$(1) + chr$(128)

Il comando, ovviamente, deve essere impartito attraverso il canale 15 del drive:

I NUOVI UNTORI

Ci siamo accorti del virus descritto in queste pagine quasi per caso: caricando un videogame di provenienza sospetta abbiamo notato che la testina del drive (che facciamo, di solito, funzionare privo del coperchio, per favorire al massimo la ventilazione) si spostava senza motivo in una zona inconsueta. Dopo aver constatato la successiva, totale mancanza di funzionamento del 1541, abbiamo disassemblato il videogame in oggetto individuando il comando relativo all'attivazione del sabotaggio.

Ecco quindi la decisione di pubblicare un "antidoto" per un inconveniente che, all'apparenza, sembra provocare danni hardware irreparabili; l'occasione, ovviamente, è utile anche per vedere da vicino il funzionamento del nostro amato drive 1541.

Se notate difficoltà di caricamento, o lampeggio del led, leggete questo articolo

```
open 1, 8, 15
print #1, comando per il drive
close1
```

Ora sappiamo dove e come scrivere nella Ram del 1541, ma che valori dobbiamo trascrivere?

La locazione 6 contiene la traccia dove si desidera posizionare la testina; nella locazione 0 dobbiamo invece inserire il codice opportuno per portare la testina sulla traccia desiderata. Il codice usato è 176, che posiziona la testina anche se la traccia non è stata formattata.

E' importante che si provveda dapprima a scrivere il valore della traccia da raggiungere e poi il comando "muovi traccia". Il DOS, infatti, esegue il comando non appena questo viene trascritto nella locazione 0, senza aspettare che venga immesso il numero di traccia nella locazione 6.

IL PROGRAMMA

Passiamo ora ad esaminare il programma, ampliamente commentato, in cui sono inserite le routine in grado di svolgere i vari compiti. Quando viene lanciato, il programma presenta un menu dotato di 4 scelte, da selezionare digitando la lettera corrispondente; appena la funzione richiesta viene eseguita, il programma ripresenta il menu iniziale.

Le opzioni sono le seguenti:

"A" :comando per posizionare la testina in corrispondenza della traccia "n"

Serve per spostare la testina su una qualunque traccia, il cui valore sia compreso tra 1 e 35 (cioè una delle tracce standard).

Da notare che questo comando, come gli altri da impartire al drive, non può funzionare se la periferica risulta bloccata in base alla procedura che tra breve descriveremo.

"B" :comando per bloccare il drive

ATTENZIONE! Con questo comando la testina viene portata sulla traccia 41; subito dopo verrà richiesto lo spegnimento, o il sempice reset, del drive; in tal modo questo si bloccherà fino a quando non userete il comando "C" per sbloccarlo nuovamente.

Se non desiderate bloccare la periferica, non spegnetela ma, mediante il comando "A", portate la testina sulla traccia 18.

E' bene sottolineare che, in caso di effettiva attivazione della scelta "B", il "sabotaggio" avrà compiuto il suo micidiale effetto fino in fondo. Non sarà quindi possibile eliminare il danno nemmeno spegnendo e riaccendendo il 1541: nè sarà possibile, soprattutto, salvare o caricare alcun programma, nemmeno quello che consente di eliminare il danno stesso! Se disponete di un registratore a cassette, quindi, vi consigliamo di utilizzarlo per registrare su nastro il programma di queste pagine in modo da non esser costretti a digitarlo nuovamente in caso di necessità.

"C" :comando sblocca drive

Questo comando, come avrete intuito, sposta la testina dalla traccia 41 (in cui si presuppone sia stata posizionata), nuovamente nella zona standard e, per di più, inizializza il drive ponendola sulla traccia 18.

```
100 REM BLOCCA SBLOCCA DRIVE 1541
110 REM BY FRANCO RODELLA
120 :
130 TR=18:REM TRACCIA INIZIALE
140 POKE53280,0:REM  BORDO NERO
150 POKE53281,11:REM SFONDO GRIGIO SCURO
160 OPEN1,8,15:REM      APRE IL CANALE DI COMUNICAZIONE CON IL DRIVE
170 GOSUB 550:REM SPOSTA LA TESTINA
180 PRINTCHR$(147)"****************"
190 PRINT"* BLOCCA DRIVE *"
200 PRINT"* @ SYSTEMS    *"
210 PRINT"* EDITORIALE   *"
220 PRINT"****************"
230 PRINT:PRINT"IL DRIVE HA LA TESTINA SULLA TRACCIA:"
240 PRINT TR
250 PRINT"A. SPOSTA LA TESTINA DOVE VUOI"
260 PRINT"B. BLOCCA IL DRIVE"
270 PRINT"C. SBLOCCA IL DRIVE"
280 PRINT"D. FINE"
290 PRINT"PREMI A , B , C , D"
300 GET A$:IF A$<"A" OR A$>"D" THEN 300:REM ACCETTA UN COMANDO
310 PRINT"■";
320 ON ASC(A$)-64 GOSUB350,420,480,340:REM ESEGUE 1A ROUTINE A 2A DEL COMANDO
330 GOTO 180
340 CLOSE1:END
350 REM SPOSTA TESTINA DOVE VUOI
360 PRINT"■■■COME VALORE DI TRACCIA POTETE INSERIRE"
370 PRINT"■UN NUMERO COMPRESO FRA 1 E 35."
380 INPUT"■■■CHE TRACCIA";A
390 IF A=0 THEN 180
400 IF A<1 OR A>35 THEN PRINT"■■■■VALORE ERRATO:":GOTO 360
410 TR=A:GOTO 550
420 TR=41:GOSUB 550:REM BLOCCA DRIVE
430 PRINT "SPEGNI IL DRIVE.":PRINT
440 PRINT "ATTENZIONE! IL DRIVE SI BLOCCHERA'!!"
450 PRINT "■PREMI UN TASTO QUANDO LO HAI RIACCESO"
460 GET A$:IF A$="" THEN 460
470 RETURN
480 FORI=1TO5:REM RIPOSIZIONA LA TESTINA
490 :FORJ=1TO4
500 ::PRINT#1,"M-W"+CHR$(0)+CHR$(28)+CHR$(1)+CHR$(216-J)
510 :NEXT J
520 NEXT I
530 PRINT#1,"I":TR=18:REM INIZIALIZZA
540 RETURN
550 PRINT#1,"M-W"+CHR$(6)+CHR$(0)+CHR$(1)+CHR$(TR):REM  CHE TRACCIA
560 PRINT#1,"M-W"+CHR$(0)+CHR$(0)+CHR$(1)+CHR$(176):REM SPOSTA LA TESTINA
570 RETURN
580 END
```

Registrate il listato anche su cassetta: non si sa mai!

"D" :fine

Al termine delle operazioni è possibile "uscire" dal programma chiudendo tutti i file precedentemente aperti.

Ricordiamo nuovamente che, uscendo dal programma dopo aver attivato la fase del blocco del drive, non sarete in grado di leggere o scrivere file; si consiglia caldamente, quindi, di sbloccare il 1541 prima di uscire.

Teniamo a precisare che nessuna responsabilità può essere addebitata alla Systems Editoriale per eventuali danni arrecati al computer e/o alla periferica nel caso di errato funzionamento o digitazione del programma; questo può essere usato dal lettore a suo esclusivo rischio e pericolo.

Per "liberare" un drive bloccato può esser sufficiente, in molti casi, impartire il comando "I"; così facendo, però, la testina del drive batterà contro il registro meccanico posto in corrispondenza della traccia 0.

Per evitare tale inconveniente si agisce in modo che la testina si sposti entro la zona standard utilizzando la locazione $1C00 del drive, che controlla direttamente la testina.

Nella malaugurata ipotesi, pertanto, che spegniate il computer (e non abbiate provveduto a registrare il programma su nastro) oppure il programma non funzioni a causa di una sua errata trascrizione, non spaventatevi: basterà riaccendere il computer e digitare:

open 1, 8, 15, "I": close 1

Con questo si reinizializza il drive, pur se in maniera (in alcuni casi) piuttosto brusca.

UNO SCROLLING IN SCATOLA DI MONTAGGIO

Come dotare i vostri programmi di una presentazione professionale

di **Lorenzo Emilitri**

In seguito all'interesse dimostrato dai nostri lettori, indichiamo, in questo articolo, alcuni trucchi del mestiere per realizzare presentazioni con effetti professionali e, comunque, "spettacolari".

Vista l'importanza dell'argomento, abbiamo scelto di cominciare con l'applicazione più classica dei Raster Interrupt, ovvero lo Smooth Scrolling orizzontale.

La procedura descritta in queste pagine vi permetterà di far scorrere all'infinito un testo scritto con Easy Script (memorizzato a parte su un dischetto) mentre "gira" un qualsiasi altro vostro programma Basic.

PRINCIPI DI BASE

Come tutti ormai sapranno, nell'integrato VIC II è presente un registro (e per la precisione quello allocato in $D016) che si occupa, tra l'altro, del cosiddetto "shifting laterale".

Probabilmente molti di voi ne conoscevano l'esistenza, ma non si sono mai posti il problema del suo funzionamento. La documentazione ufficiale Commodore, a meno che non si possegga la "Guida di Riferimento per il Programmatore", come al solito brilla per chiarezza, visto che a pagina 154 del manuale del C/64 (edizione Italiana) il registro in questione non è nemmeno(!) dotato di un misero commento, e le sigle riportate nella tavola illustrativa, da sole, non servono ad altro che a confondere le idee al programmatore che abbia intenzione di utilizzarlo.

Fortunatamente Commodore Computer Club (cioè, modestamente, noi) ha deciso di portare luce su un argomento così importante, cosicchè i nostri lettori, famosi in tutto l'universo conosciuto per la loro intelligenza superiore (non per niente scelgono C.C.C.), possano finalmente adottare tale tecnica nei propri programmi.

Premettiamo che se le informazioni tecniche qui riportate possono essere sfruttate al meglio solo da coloro che hanno già una certa conoscenza della macchina, il programma può essere utilizzato anche dai principianti che hanno appena acquistato un C/64. Se, quindi, appartieni a questi ultimi tralascia pure il resto del paragrafo e passa direttamente alla digitazione del programma pubblicato.

Anche i principianti possono usare la procedura descritta in queste pagine

REGISTRI

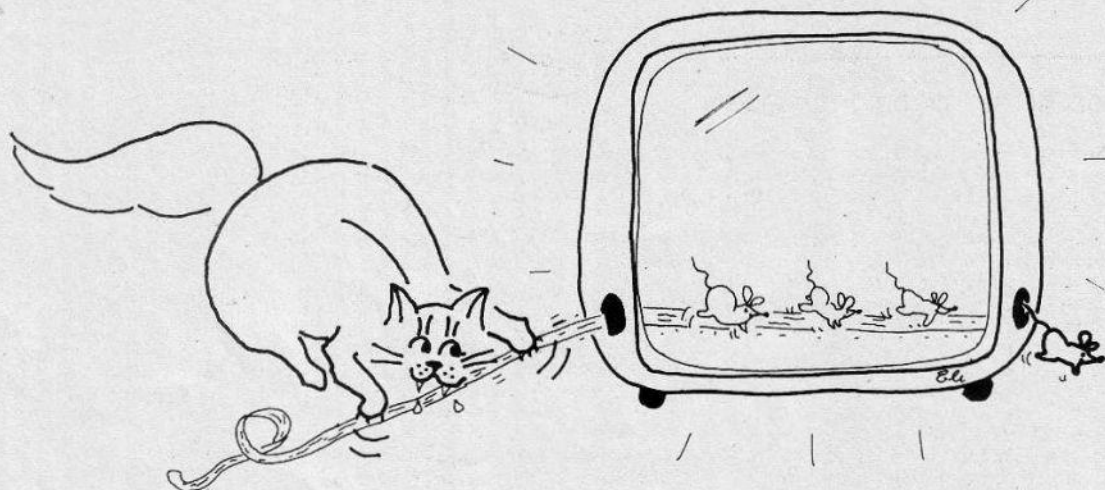
Il registro $D016 (decimale 53270) ha, tra le altre sue funzioni, quella di gestire lo shifting laterale. Ciò significa che, impostando un opportuno valore nei suoi 3 bit meno significativi, è possibile, sia pur di poco, variare la posizione della zona visibile dello schermo.

Come si può intuire abbastanza facilmente, per realizzare uno scrolling laterale è sufficiente:

- Decrementare il contenuto dei tre bit meno significativi fino a che contengano zero;

- Spostare tutti i caratteri a sinistra di una posizione, eliminando il primo della fila ed inserendone un altro al quarantesimo posto.

E' inoltre facilmente intuibile che, per realizzare un movimento effettivamente fluido, bisogna ricorrere ai Raster Interrupt in modo che tutte le operazioni siano svolte in maniera sincrona allo schermo stesso.

Giovedini giunge presso l'aerorifornitore ...

LA FAMIGLIA SCROLL

Leggendo alcuni articoli molti di voi si saranno scontrati con un gergo quantomeno inusuale. Per rendere più facile la comprensione dell'articolo, ecco un breve glossario.

SCROLLING
Il testo che viene fatto passare sul video da una routine apposita (vedi articolo). Si definisce SMOOTH quando non procede a scatti; ROUGH in caso contrario.

SCROLLER
La routine che si occupa di far "scrollare" (brutto verbo, anzi orribile, ma rende l'idea) un testo. Si distingue fra SINGLE e MULTI SPEED SCROLLER; il primo tipo (come quello riportato nell'articolo) ha una sola velocità di scorrimento dei caratteri, mentre molti altri scrollers hanno una velocità variabile a piacere dall'utente inserendo certi caratteri nel testo.

SCROLLIE
(Si trova anche la forma SKROLLIE): Diminutivo di Scrolling. Nella terminologia usata dagli Hackers, lo stesso che Scrolling.

Nonostante tutto ciò, se qualcuno decide di lanciarsi sull'assemblatore per scrivere il proprio Super - Mega - Scroller - Multi - Velocità - Multi - Color, ebbene egli va sicuramente incontro ad una delusione di carattere prevalentemente estetico: il bordo si sposterà a scatti insieme ai caratteri.

"Come? -diranno tutti coloro che ci hanno seguito- Allora quelle sopra sono tutte chiacchiere inutili?"

No di certo, ma esiste un altro bit nel registro $D016, e per la precisione il numero 3, che si incarica di abilitare il modo 39 colonne, essenziale se si

GLI ALTRI WORD PROCESSOR

Se non possedete Easy Script, oppure siete affezionati al vostro vecchio Word Processor, niente paura: è sufficiente avere un po' di familiarità col Basic per adattare il programma qui presentato a lavorare anche con i file creati dal vostro amato Word Processor.

E' sufficiente ricordare che:

1) Se i file salvati sono di tipo PRG dovete effettuare due istruzioni GET# a vuoto, per evitare che i due byte che codificano l'indirizzo d'inizio del testo finiscano nel testo stesso, con dubbi risultati dal punto di vista grafico.

2) Se i file sono in Codici Video (D/Codes) dovete eliminare le linee da 265 a 285.

3) Se i file sono in PET ASCII non apportate alcuna modifica alle linee 265 - 285;

4) Se il vostro Word Processor usa invece altre codifiche (vero ASCII, tanto per citarne una) dovete modificare opportunamente le linee 265 - 285.

Ad esempio, se utilizzate SpeedScript dovreste:

A) Inserire una linea contenente due GET# a vuoto...

246 GET #8, A$: GET #8, A$

Quindi modificare come segue la riga 240, per evitare che in fase di apertura del file venga emesso dal drive un errore, visto che i file di Easy Script sono SEQuenziali e quelli di Speedscript PRoGramma...

240 OPEN 8, 8, 8, NS$ + ",P ,W"

B) Infine cancellare le linee 265 - 285 visto che SpeedScript usa direttamente codici video e non il solito PET ASCII.

Nonostante tutto, vi rinnoviamo l'invito a procurarvi e ad usare Easy Script: tale w/p è infatti (a nostro parere, che è Cassazione) il miglior Word Processor oggi disponibile sul C/64.

vuole che il proprio scrolling fluisca in modo ACE (senza Strapp) da un bordo all'altro dello schermo.

IL PROGRAMMA

Dopo il discorso introduttivo, siamo finalmente pronti per scrivere una routine di Smooth Scrolling perfettamente funzionante (i più pigri possono benissimo limitarsi ad osservare il programma ed il disassemblato qui riportati, anche se niente insegna come la pratica).

Il programma è stato realizzato in modo da essere utilizzabile anche dal novizio meno esperto. E' sufficiente possedere una copia di Easy Script (Come non l'avete? E cosa aspettate a procurarvela, magari da qualche amico?) o di un altro Word Processor, un'unità a dischi ed un po' di pazienza.

COME USARE IL PROGRAMMA

Esso supera i limiti di tutti gli altri programmi del suo tipo, visto che:

- Crea file completamente compatibili col Basic, fatti apposta per girare contemporaneamente ad esso;

- Usa l'ultima linea di schermo e non, come molti programmi similari, la prima, spesso utilizzata dal programmatore per disegni, menu o diavolerie simili;

- Non interferirà mai nè con l'output dei vostri programmi ne con quello del Basic (LISTare per credere).

Dopo averlo digitato, salvato e mandato in esecuzione, vi chiederà il nome del file sorgente e di quello oggetto. Come i più acuti avranno intuito, il nome del file sorgente è quello di un file salvato con Easy Script (se voi, nonostante tutto, siete affezionati al vostro vecchio Word Processor e vorrete utilizzarlo, leggete il riquadro apposito per le spiegazioni su come adattare il programma). Il nome del file oggetto, ovviamente, sarà il nome della routine (com-

La manipolazione del Raster Register offre sempre elaborazioni suggestive

```
100 REM * * * * * * * * * * * * * * * * * *
105 REM * SMOOTH SCROLL MAKER * V 1.0 *
110 REM * BY LORENZO EMILITRI * 01/89 *
115 REM * * * * * * * * * * * * * * * * * *
120 :
125 REM ALLOCA ROUTINE IN L/M
130 FOR A = 49152 TO 49337
135 : READ B : C = C+B :POKE A,B
140 NEXT A
145 IF C <> 28035 THEN PRINT"DATA ERROR!" : STOP
150 REM DEFINISCE FUNZIONI BYTE LO/HI
155 DEF FNHI(X) = INT(X/256)
160 DEF FNLO(X) = X-FNHI(X)*256
165 REM SHIFT INIZIALE: 0
170 POKE 49338,200
175 REM INIZIO DEL TESTO: 49383
180 POKE 49341,231
185 POKE 49342,192
190 REM PULISCE BUFFER
195 GOSUB 450
200 REM CARICAMENTO DEL TESTO
205 POKE 53280,0: POKE 53281,0
210 PRINT CHR$(147) TAB(5) CHR$(30);
215 PRINT "CARICAMENTO DI UN TESTO IN RAM"
220 PRINT : PRINT : INPUT "NOME SORGENTE ";NS$
225 INPUT "NOME OGGETTO  ";NO$
230 IF NS$="" OR NO$="" THEN GOTO 200
235 OPEN 15,8,15,"I0"
240 OPEN 8,8,8,NS$+",S,R"
245 GOSUB 475
250 T=0
255 GET#8,A$ : A=ASC(A$)
260 IF ST<>0 THEN 295
262 IF A=13 THEN A=32:GOTO290
265 IF A<32 THEN GOTO 255
270 IF A>63 AND A<96 THEN A=A-64
275 IF A>95 AND A<128 THEN A=A-32
280 IF A>160 AND A<192 THEN A=A-64
285 IF A>191 AND A<255 THEN A=A-128
290 POKE 49383+T,A : T=T+1 : GOTO 255
295 CLOSE 8
300 GOSUB 475
305 CLOSE 15
310 PRINT : PRINT"PREMI UN TASTO PER VEDERE LO SCROLLING."
315 GET A$: IF A$="" THEN GOTO 315
320 POKE 49339, FNLO(49383+T)
325 POKE 49340, FNHI(49383+T)
330 SYS 49152
335 PRINT "PREMI UN TASTO PER SALVARE LO SCROLLING."
340 GET A$: IF A$="" THEN GOTO 340
```

Annullando la gravità, Primo evita di precipitare e riesce a raggiungere la locazione $0400, dove si trova la sua portaerei.
Nello stesso istante, un altro F18 con a bordo un altro Primo Giovedini si prepara a rientrare...

```
345 REM SAVE
350 REM DISABILITA RASTER INTERRUPTS...
355 POKE 53274,0
360 REM ...PULISCE BUFFER VIDEO...
365 GOSUB450
370 REM ...E SALVA
375 OPEN 15,8,15,"I0"
380 OPEN 8,8,8,NO$+",P,W"
385 GOSUB 475
390 PRINT#8, CHR$(0) CHR$(192);
395 FOR A = 49152 TO 49383+T
400 : PRINT#8,CHR$(PEEK(A));
405 NEXT A
410 CLOSE 8
415 GOSUB 475
420 CLOSE 15
425 REM RIABILITA RASTER IRQ
430 POKE 53274,1
435 REM FINE PROGRAMMA
440 END : END : END : END : END : END
445 :
450 REM SUBRO: PULISCE BUFFER VIDEO
455 FOR A = 0 TO 39
460 : POKE 49343+A,32
465 NEXT A
470 RETURN
475 REM SUBRO:INTERROGA CANALE D'ERRORE
480 INPUT#15,EN,ES$,ET,ES
485 IF EN=0 THEN RETURN
490 PRINT : PRINT "DISK ERROR:"
495 PRINT EN,ES$,ET,ES
500 STOP
505 :
510 REM DATA L/M (VEDI DISASSEMBLATO)
515 DATA 120,169,053,141,020,003,169
520 DATA 192,141,021,003,169,219,141
525 DATA 018,208,173,017,208,041,127
530 DATA 141,017,208,169,000,141,014
535 DATA 220,169,001,141,026,208,088
540 DATA 173,187,192,056,233,001,133
545 DATA 251,173,188,192,133,252,176
550 DATA 002,198,252,096,173,025,208
555 DATA 141,025,208,032,164,192,160
560 DATA 000,200,192,005,208,251,173
565 DATA 186,192,141,022,208,206,186
570 DATA 192,160,000,234,200,192,053
575 DATA 208,250,032,164,192,173,186
580 DATA 192,201,191,240,003,076,156
585 DATA 192,169,199,141,186,192,160
590 DATA 001,185,191,192,136,153,191
595 DATA 192,200,200,192,040,208,243
600 DATA 230,251,208,002,230,252,165
605 DATA 252,205,188,192,208,017,165
610 DATA 251,205,187,192,208,010,173
615 DATA 189,192,133,251,173,190,192
620 DATA 133,252,160,000,177,251,141
625 DATA 230,192,169,200,141,022,208
630 DATA 076,049,234,160,000,185,192
635 DATA 007,072,185,191,192,153,192
640 DATA 007,104,153,191,192,200,192
645 DATA 040,208,237,096
650 END
```

prensiva di testo) che verrà salvata su disco.

Appena il computer avrà ricevuto i due dati, il drive inizierà a ronzare e, dopo un tempo direttamente proporzionale alla lunghezza del file di testo stesso (abbiate pazienza: lavora in Basic!), apparirà lo scrolling. Se tutto funziona a dovere premete la barra spaziatrice per salvare la routine; se, per qualsiasi motivo non voleste salvare (errori nel testo o altro) premete semplicemente RUN / STOP.

Ora è presente sul vostro disco una routine di Smooth Scrolling perfettamente funzionante. Per utilizzarla nei vostri programmi, caricatela con...

LOAD "nome", 8, 1
NEW

...e fatela partire con...

SYS 49152

Per chiarire le idee, supponiamo che il file generato da Easy Script (già presente su disco) abbia nome "Prova" e che, dopo l'applicazione della procedura descritta, sia stato salvato, come file oggetto, con il nome "Prova smooth" sullo stesso disco. Per vedere scorrere all'infinito il testo mentre gira un vostro programma, digitate quest'ultimo nel modo seguente:

```
100 rem esempio di smooth scroll
105 :
110 if a=1 then 125
120 a=1: load "prova smooth", 8, 1
125 sys 49152
130 input "primo addendo"; a
140 input "secon.addendo"; b
150 print "la somma vale: " a + b
160 print: print: goto 130
```

E' ovvio che l'esempio è banale, ma i più furbi avranno capito come utilizzare correttamente l'intera procedura.

Un qualsiasi testo (lungo anche qualche Kbyte) scorrerà all'infinito sul video

```
; ********************************
; * smooth scrolling orizzontale *
; *  nell' ultima riga di testo  *
; * senza interferire col basic! *
; *------------------------------*
; *    by   lorenzo emilitri     *
; ********************************
;          versione 1.03
;
;---------------------------------
;       routine preparatoria
;---------------------------------
        *=$c000
        sei           ;disabilita irq.
        lda #<main    ;punta l'irq
        sta $0314     ;a 'main'.
        lda #>main    ;
        sta $0315     ;
        lda #$db      ;raster interrupt
        sta raster    ;alla riga $db.
        lda hirast    ;resetta bit piu'
        and #$7f      ;significativo
        sta hirast    ;del raster reg.
        lda #00       ;disab. timer 'a'
        sta timer     ;del cia #1.
        lda #01       ;abilita raster
        sta vicirq    ;interrupts.
        cli           ;riabilita irq.
        lda lfine     ;togli al valore
        sec           ;contenuto nella
        sbc #01       ;coppia lfine e
        sta $fb       ;hfine un'unita'
        lda hfine     ;quindi forzalo
        sta $fc       ;in $fb/$fc.
        bcs exit      ;
        dec $fc       ;
exit    rts           ;ed esci.
;----------------------------------
;     routine per smooth scroll
;----------------------------------
main    lda vicflg ;conferma raster
        sta vicflg ;interrupt.
        jsr move   ;esegui 'move'.
        ldy #00    ;breve ciclo di
loop4   iny        ;ritardo.
        cpy #05    ;
        bne loop4  ;
        lda tshift ;forza il valore
        sta shift  ;di 'tshift' nel
                   ;reg.per lo shift
                   ;laterale.
        dec tshift ;decrementalo.
        ldy #00    ;altro breve loop
loop1   nop        ;di ritardo.
        iny        ;
        cpy #$35   ;
        bne loop1  ;
        jsr move   ;esegue 'move'.
        lda tshift ;se ha shiftato
        cmp #$bf   ;di 8 posizioni
        beq scrl   ;salta a scrl.
        jmp out    ;se no, va a out.
scrl    lda #$c7   ;shift di 8.
        sta tshift ;
        ldy #$01   ;sposta tutti i
loop2   lda buff,y ;caratteri nel
```

"DENTRO" IL PROGRAMMA BASIC

Il programma Basic, come si può facilmente realizzare dando una breve occhiata al listato qui pubblicato, si può idealmente dividere in 8 parti:

125 - 145 Carica routine in L/M in Ram;

150 - 195 Setta i suoi parametri (Buffer, inizio del testo, posizione del testo all'inizio);

200 - 305 Carica il testo in Ram (dalla linea 265 alla linea 285 si trova la routine di conversione da codici ASCII a Display / code, ovvero codici video);

310 - 330 Attiva la routine in Linguaggio Macchina allocata a partire da $C000 per vedere il testo;

335 - 430 Salva la routine in L/M su disco.

Le ultime due parti del programma sono due subroutine (Niklaus Wirt docet):

450 - 470 Pulisce (= riempe con spazi) il buffer usato dal programma;

475 - 500 Interroga il canale d'errore del drive.

Volendo modificare la routine di scrolling stessa, si tengano presenti alcune locazioni fondamentali per il suo corretto uso:

```
       dey          ;buffer a sinist.
       sta buff,y ;di 1 posto
       iny          ;
       iny          ;
       cpy #40      ;
       bne loop2    ;
       inc $fb      ;increm. $fb/$fc.
       bne skip     ;
       inc $fc      ;
skip   lda $fc      ;se e' arrivato
       cmp hfine    ;al valore della
       bne out2     ;coppia 'lfine'-
       lda $fb      ;-'hfine'....
       cmp lfine    ;
       bne out2     ;
       lda liniz    ;allora torna da
       sta $fb      ;capo.
       lda hiniz    ;
       sta $fc      ;
out2   ldy #00      ;mette la nuova
       lda ($fb),y;lettera in 40'
       sta buff+39;posizione.
out    lda #$c8     ;rimette a posto
       sta shift    ;il video.
       jmp irq      ;e salta a irq.
;---------------------------------
;  subroutine move:  scambia la
;  ultima riga di schermo con il
;  contenuto del buffer.
;---------------------------------
move   ldy #00      ;contatore=0.
loop3  lda rg24,y ;scambia i
       pha          ;valori contenuti
       lda buff,y ;a partire dalla
       sta rg24,y ;ultimma riga con
       pla          ;quelli contenuti
       sta buff,y ;nel buffer.
       iny          ;
       cpy #40      ;
       bne loop3    ;
       rts          ;ed esce.
;
;---------------------------------
;              simboli:
;---------------------------------
raster =$d012
hirast =$d011
timer  =$dc0e
vicirq =$d01a
vicflg =$d019
shift  =$d016
irq    =$ea31
rg24   =$07c0
;---------------------------------

;            variabili:
;---------------------------------
tshift  .byte $c8
lfine   .byte $00 ;fine:    $c500
hfine   .byte $c5
liniz   .byte $00 ;inizio: $c300
hiniz   .byte $c3
buff    .byte '          '
        .byte '          '
        .byte '          '
        .byte '          '
```

49338 Valore iniziale di shifting (quello che per primo verrà forzato in $D016);

49339 / 49340 Indirizzo di fine del testo (+1);

49341 / 49342 Indirizzo d'inizio del testo;

49343 - 49382 Buffer usato dal programma;

49383 - in poi: Testo dell'utente.

PER FINIRE...

La routine di Smooth Scrolling, per quanto possa essere utile, non ha assolutamente alcuna pretesa di impressionare l'utente, specie se smaliziato, con prestazioni grafiche ad altezza veramente professionale.

Visto che il tema dello Smooth Scrolling di caratteri è già abbastanza complicato di per sè, per non appesantire troppo la trattazione si è deciso di non includere in questo articolo anche uno scroller "State - of - the - Art" (= il migliore oggi realizzabile sul C/64) con caratteri che scorrono sul bordo dello schermo ed effetti raster di sottofondo.

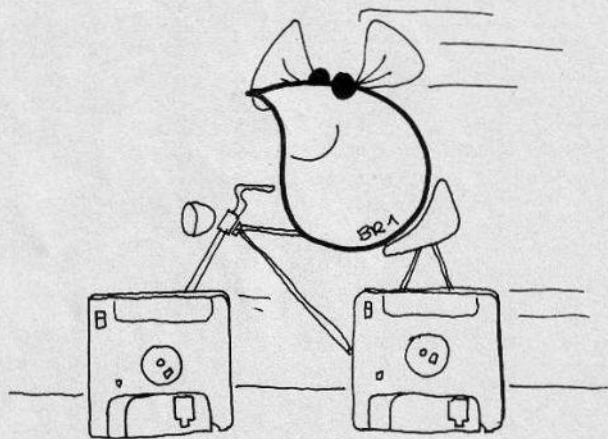
Ciò non significa però che C.C.C. non tratterà questo tema; anzi, ne approfittiamo per lanciare una sfida agli smanettoni più appassionati: provate a realizzare uno scrolling come quello descritto sopra.

P.S:

L'importante non è vincere, mâ partecipare!

UN SINGOLARE INSEGUIMENTO

Una divertente applicazione dell'interrupt sul C/64

di **Fabio Sorgato**

Ancora un'applicazione dell'interrupt!

Di interrupt abbiamo parlato innumerevoli volte, non tanto perchè questo argomento ci piaccia in modo particolare, quanto per la versatilità delle sue applicazioni e per l'importanza che assume nel funzionamento del C/64.

Ripetiamo tuttavia, per chi non lo sapesse ancora, che l'interrupt è una "chiamata" che il microprocessore esegue automaticamente, tramite un salto indiretto alle locazioni $FFFE - $FFFF della memoria, 60 volte al secondo in condizioni normali. In tal modo viene interrotta qualsiasi operazione il computer stia facendo in quel momento (onde il nome interrupt = interruzione, sospensione) e viene invece eseguita la routine "puntata" dalle suddette celle di memoria; questa non dovrà terminare con un'istruzione RTS (ReTurn from Subroutine), ma con un RTI (ReTurn from Interrupt).

Le locazioni $FFFE e $FFFF della Rom del C/64 rinviano, in condizioni standard, alla prima parte della routine di interrupt che risiede a partire dalla locazione $FF48 e che si occupa di salvare l'accumulatore, i registri X ed Y nello stack, per evitare che subiscano modifiche durante la seconda parte della routine, e di decidere se la richiesta di interrupt venga dal sistema o sia stata forzata da un'istruzione BRK. Nel caso dell'interrupt di sistema viene eseguito un salto indiretto alle locazioni $0314 - $0315, che puntano a $EA31, dove risiede la "continuazione" della routine di interrupt.

Questa parte della routine si occupa di incrementare l'orologio, di fare lampeggiare il cursore, di controllare il motore del registratore, di scandire la tastiera ed infine di ripristinare i registri e l'accumulatore nonchè di eseguire RTI.

Modificando i puntatori $0314 - $0315 si può fare eseguire una routine ogni IRQ (richiesta di interrupt), nel caso si tratti di una routine che debba convivere con il Basic è necessario che questa non termini con un RTI, ma con con un salto a $EA31 o che almeno si occupi della scansione di tastiera e del ripristino di A, X e Y, pena non potere più impostare alcun comando, non funzionando neppure run / stop e restore.

UNO SPRITE A SPASSO PER LO SCHERMO

Viene qui pubblicato, oltre al programma Basic, il corrispondente disassemblato commentato della

```
10 REM **
20 REM ** UN SINGOLARE INSEGIMENTO
30 REM **   ...DIGITA E VEDRAI !
40 REM **
50 REM ** (GIOCO PER C/64)
60 REM **
70 REM ** BY FABIO SORGATO
80 REM **
90 REM
100 CK=0:FOR K=49152 TO 49376
105 READ A:CK=CK+A:POKE K,A:NEXT
107 IF CK<>28166 THEN PRINT"ERRORE NEI DATI LM":END
110 FORK=896 TO 1023:POKE K,0:NEXT
115 CK=0
120 FORK=0 TO 33
125 READ A:POKE 896+K,A
127 CK=CK+A:NEXT
130 FORK=0 TO 30
135 READ A:POKE 960+K,A
137 CK=CK+A:NEXT
138 IF CK<>3127 THEN PRINT"ERRORE NEGLI ALTRI DATI!":END
140 POKE 53269,1:POKE 2040,15:POKE 53287,0:REM ABILITA SPRITE
150 POKE 53249,238:POKE 53264,1:POKE 53248,75:REM POSIZIONE INIZIALE
160 POKE 250,0:POKE 53275,0:POKE 53277,0
165 POKE53271,0
170 PRINT"SALVA IL CURSORE DALLA ZANZARA !"
180 SYS49152
980 :
990 REM
995 REM DATI LM
996 REM
997 :
1000 DATA 120,169,033,141,020,003,169
1010 DATA 192,141,021,003,088,096,065
1020 DATA 073,085,084,079,032,033,033
1030 DATA 000,065,065,082,071,072,072
1040 DATA 072,046,046,032,000,165,250
1050 DATA 201,255,240,027,170,208,012
1060 DATA 165,161,024,105,002,133,002
1070 DATA 230,250,076,049,234,165,161
1080 DATA 197,002,208,247,169,255,133
1090 DATA 250,208,241,173,248,007,073
1100 DATA 001,141,248,007,141,024,212
1110 DATA 165,214,010,010,010,105,050
1120 DATA 205,001,208,008,240,011,174
1130 DATA 001,208,144,002,232,232,202
1140 DATA 142,001,208,169,000,133,252
1150 DATA 165,211,201,041,144,003,056
1160 DATA 233,040,010,010,010,038,252
1170 DATA 024,105,024,133,251,144,002
1180 DATA 230,252,173,016,208,041,001
1190 DATA 197,252,240,056,144,039,174
```

```
1200 DATA 000,208,208,008,173,016,208
1210 DATA 041,254,141,016,208,206,000
1220 DATA 208,104,165,162,208,015,162
1230 DATA 000,189,013,192,240,006,157
1240 DATA 119,002,232,208,245,134,198
1250 DATA 076,051,192,238,000,208,208
1260 DATA 228,173,016,208,009,001,141
1270 DATA 016,208,208,218,165,251,205
1280 DATA 000,208,144,195,208,232,040
1290 DATA 208,226,162,000,134,250,189
1300 DATA 022,192,240,006,157,119,002
1310 DATA 232,208,245,134,198,076,051
1320 DATA 192
1321 :
1322 REM
1323 REM ALTRI DATI
1324 REM
1325 :
1330 DATA 128,128,000,065,000,000,034
1340 DATA 000,000,028,240,000,031,248
1350 DATA 000,028,112,000,042,000,000
1360 DATA 089,000,000,156,128,000,028
1370 DATA 000,000,028,000,000,008
1380 DATA 128,128,000,065,096,000,034
1390 DATA 224,000,029,192,000,031,000
1400 DATA 000,028,000,000,042,000,000
1410 DATA 089,000,000,184,128,000,112
1420 DATA 000,000,096
1430 END
```

Il semplice gioco può essere personalizzato dal lettore esperto

divertente mini-routine che si occupa di far inseguire il cursore da una fastidiosa zanzara.

Da notare come l'animazione dello sprite venga eseguita scambiando, ad ogni chiamata, le immagini della zanzara disegnata in due diverse posizioni.

Sottolineamo il fatto che non è possibile utilizzare la routine del S.O. di chrout ($FFD2) durante una chiamata di interrupt perchè la suddetta routine contiene un CLI nella parte finale; per stampare un messaggio è quindi necessario utilizzare il buffer di tastiera.

In ultimo: la locazione 56437 controlla la frequenza degli interrupt; più basso è il valore, maggiore è la frequenza degli interrupt (il valore normale è 52).

Provate a controllare l'effetto sulla routine della zanzara.

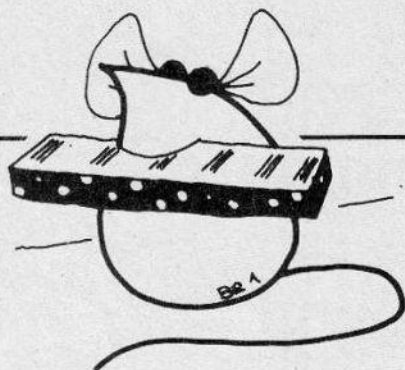
Maggiore è il numero degli interrupt, minore è la velocità di esecuzione dei programmi dato che la

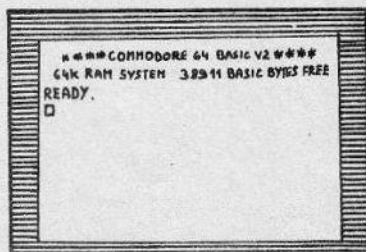
routine di interrupt viene chiamata un numero elevato di volte ogni secondo; se provate con...
POKE 56437, 0
...otterrete una velocità degna del vecchio TI-99 della Texas.

```
DISASSEMBLATO COMMENTATO

;
;Modifica i puntatori di IRQ
;
       SEI        ;Disabilita interrupt
       LDA #$21   ;e punta alla nuova
       STA $0314  ;routine di IRQ
       LDA #$C0   ;($C021).
       STA $0315  ;
       CLI        ;Riabilita interrupt;
       RTS        ;torna al basic.
;Scritte utilizzate dalla routine.
;
BC00D .BYTE 'AIUTO !!',$00
BC016 .BYTE 'AARGHHH.. ',$00

;*** Nuova routine di IRQ

;
;Decide quale routine eseguire in base
;al contenuto di $FA.
;
BC021 LDA $FA     ;Se $FA contiene 255,
      CMP #$FF    ;esegue la routine
      BEQ BC042   ;sprite move.
      TAX         ;Se il contenuto e'
      BNE BC036   ;<>0 esegue la routine
                  ;wait, altrimenti
;
;Decide il tempo d'attesa prima che lo
;sprite inizi a muoversi
;
      LDA $A1     ;legge il contenuto
      CLC         ;attuale del byte 2
                  ;del clock,
      ADC #$02    ;addiziona il tempo
                  ;d'attesa
      STA $02     ;e deposita il risul-
                  ;tato.
      INC $FA     ;Abilita la routine
                  ;di wait.
BC033 JMP $EA31   ;Continua la IRQ
                  ;routine.
;
;Routine di wait (tempo d'attesa)
;
BC036 LDA $A1     ;Legge il clock,
      CMP $02     ;se ha teminato il
      BNE BC033   ;tempo d'attesa,
      LDA #$FF    ;abilita la routine
      STA $FA     ;che muove lo sprite.
      BNE BC033   ;Continua IRQ.
;
;Routine muovi sprite.
;
BC042 LDA $07F8   ;Legge il puntatore
      EOR #$01    ;allo sprite 0, esegue
      STA $07F8   ;lo swap del bit 0,
                  ;per alternare le due
                  ;posizioni delle ali,
      STA $D418   ;copia il contenuto
                  ;nel volume per creare
                  ;il ronzio.
      LDA $D6     ;Legge la posizione
      ASL A       ;y del cursore,
      ASL A       ;moltiplica per 8,
      ASL A       ;
      ADC #$32    ;somma 50 e lo compara
      CMP $D001   ;alle y dello sprite,
      PHP         ;salvando i falg;
      BEQ BC065   ;se sono uguali conti-
                  ;nua;
      LDX $D001   ;
      BCC BC061   ;se lo sprite e' piu'
      INX         ;in alto, incrementa
      INX         ;la posizione delle y
                  ;dello sprite,
BC061 DEX         ;alrimenti le
      STX $D001   ;decrementa.
BC065 LDA #$00    ;
      STA $FC     ;
      LDA $D3     ;Legge le x del curso-
                  ;re,
      CMP #$29    ;se si trova su una
      BCC BC072   ;linea di 80 carat-
      SEC         ;teri, sottrae 40
      SBC #$28
BC072 ASL A       ;moltiplica per 8,
      ASL A       ;
      ASL A       ;
      ROL $FC     ;considerando anche il
      CLC         ;byte alto,
      ADC #$18    ;addiziona 24,
      STA $FB     ;
      BCC BC080   ;
      INC $FC     ;
BC080 LDA $D010   ;Se il byte alto delle
      AND #$01    ;x dello sprite 0 e'
      CMP $FC     ;uguale al byte alto
                  ;delle x del cursore,
      BEQ BC0C1   ;compara i byte bassi,
      BCC BC0B2   ;se e' minore incre-
                  ;menta le x dello
                  ;sprite, altrimenti
BC08B LDX $D000   ;legge il byte basso
      BNE BC098   ;della posizione x
                  ;dello sprite, se e'
                  ;uguale a 0,
      LDA $D010   ;spegne il byte alto,
      AND #$FE
      STA $D010
BC098 DEC $D000   ;decrementa le x;
BC09B PLA         ;toglie i flag dallo
                  ;stack
      LDA $A2     ;ogni 255 sessantesimi
      BNE BC0AF
      LDX #$00    ;mette copia il primo
BC0A2 LDA BC00D,X ;messaggio nel buffer
      BEQ BC0AD   ;di tastiera
      STA $0277,X
      INX
      BNE BC0A2
BC0AD STX $C6
BC0AF JMP BC033   ;ed esce.
;
BC0B2 INC $D000   ;incrementa le x dello
      BNE BC09B   ;sprite, se superano
                  ;255,
      LDA $D010   ;abilita il byte alto,
      ORA #$01    ;
      STA $D010   ;
      BNE BC09B   ;ed esce.
;
BC0C1 LDA $FB     ;Se il byte basso
                  ;delle x del cursore
                  ;e' minore delle x
      CMP $D000   ;dello sprite, decre-
      BCC BC08B   ;menta le x dello spr.
      BNE BC0B2   ;se e' maggiore,
                  ;incrementa,
      PLP         ;altrimenti richiama
                  ;i flag della compara-
                  ;zione delle y;
      BNE BC0AF   ;se le y erano diver-
                  ;se, esce.
      LDX #$00    ;Abilita la routine
                  ;che decide il tempo
                  ;d'attesa,
      STX $FA
BC0D1 LDA BC016,X ;copia nel buffer
      BEQ BC0DC   ;di tastiera il
      STA $0277,X ;secondo messaggio
      INX
      BNE BC0D1
BC0DC STX $C6
      JMP BC033   ;ed esce.
```

```
*-$c000
       sei
       lda #<newin
       sta $0314
       lda #>newin
       sta $0315
       cli
       rts
msgx   .byte 'aiuto !!',0
xmsg2  .byte 'aarghhh.. ',0
newin  lda $fa
       cmp #$ff
       beq abil
       tax
       bne contr
       lda $a1
       clc
       adc #$02
       sta $02
       inc $fa
exit   jmp $ea31
contr  lda $a1
       cmp $02
       bne exit
       lda #$ff
       sta $fa
       bne exit
abil   lda 2040
       eor #$01
       sta 2040
       sta 54296
       lda $d6
       asl a
       asl a
       asl a
       adc #$32
       cmp $d001
       php
       beq abila
       ldx $d001
       bcc abil3
       inx
       inx
abil3  dex
       stx $d001
abila  lda #$00
       sta $fc
       lda $d3
       cmp #41
       bcc abilx
       sec
       sbc #40
abilx  asl a
       asl a
       asl a
       rol $fc
       clc
       adc #$18
       sta $fb
       bcc abil4
       inc $fc
abil4  lda $d010
       and #$01
       cmp $fc
       beq abil2
       bcc incx
decx   ldx $d000
       bne abil7
       lda $d010
       and #$fe
       sta $d010
abil7  dec $d000
exi2   pla
       lda $a2
       bne exi3
       ldx #$00
exiy   lda msgx,x
       beq exix
       sta 631,x
       inx
       bne exiy
exix   stx 198
exi3   jmp exit
incx   inc $d000
       bne exi2
       lda $d010
       ora #$01
       sta $d010
       bne exi2
abil2  lda $fb
       cmp $d000
       bcc decx
       bne incx
       plp
       bne exi3
       ldx #$00
       stx $fa
jjjjj  lda xmsg2,x
       beq kkkk
       sta 631,x
       inx
       bne jjjjj
kkkk   stx 198
       jmp exit
```

Beh, anche con i nomi in "Default", Primo Giovedini non finisce mai di stupire ...!
Ad ogni modo, tutto è bene quel che finisce bene: l'emulatore di Primo non era altro che suo fratello, ed ora i due Giovedini lavoreranno insieme per il "CRAY-ONE EMULATOR" !!!

Un momento! Prima di congedarci, vorrei informarvi che la schermata di Reset del "64" è stata in realtà ottenuta da un VIC-20 che emulava un Amiga che emulava uno Spectrum che emulava un "128" che emulava un Plus-4 che emulava ... ...

VUOI PUBBLICARE UN ANNUNCIO GRATUITO?

Commodore Computer Club ti offre la possibilità di pubblicare GRATUITAMENTE il tuo annuncio riguardo scambio o vendita di software, vendita o acquisto di apparecchi usati, ricerca di amici per fondare un club e così via. Utilizza lo spazio vuoto, incorniciato qui a fianco, usando un pennarello e tenendo presente che IL TUO STESSO ANNUNCIO verrà pubblicato così come ci perviene in Redazione, senza alcuna modifica. Ti consigliamo, pertanto, di usare una calligrafia chiara, a stampatello e dotata di caratteri sufficientemente grandi per consentire un'agevole lettura. Avranno precedenza gli annunci più curati graficamente e quelli di "carattere" marcatamente hobbystico. Invia l'annuncio in busta chiusa, affrancata secondo le norme vigenti, indirizzando a:

Commodore Computer Club
Viale Famagosta, 75
20142 MILANO

Scambio per C.64 qualsiasi tipo di programma. Disposto anche per l'acquisto Leonelli Davide loc. Diaccioni 25/15 57025 Piombino (LI) tel. (0565) 40990

Vendo adattore telematico 6499 per C.64-128 tel. 085 - 692792 (solo di mattina) chiedere di Leonardo

Gaeta Computers 1982 software - for C.64/128 - Amiga Via Tevere, 31 - Rozzano (MI) Cap. 20089 tel 02/8258192 Gennaro Gaetano

Mi chiamo Federico - ho 11 anni - cerco amico coetaneo con PC Compatibile per "scambi programmi " tel. 06/6811036 - ore 17-20

Cerco utenti per adattatore telematico scopo: scambio di programmi, giochi, informazioni ecc... A chi fosse interessato telefoni allo 090/710756. Donnina Ivan Via Udine, 4 Messina

Vendo per C.64 programmi (ultime unità) a 300 lire a disco (questo compreso) /e non posso mandare liste game). Per ulteriori informazioni telefonare allo 06/4387276 Marco Stefanucci - Via dei Crispolti 78 00158 Roma (RM)

Cerco stampante MPS 803 o Compatibile - Possibilmente nella prov. di venezia tel. 041/641179 dopo le ore 18 - Ermanno

Vendo giochi e programmi di qualsiasi tipo per Amiga-500, e giochi per 64 su disco e cassetta. Alessio Mariano - Via Baldissera 5/3 Mestre (VE) tel. 041/5341727

Vuoi fare un affare? Vendo Commodore 64 tastiera, registratore, disc drive 1541, porta dischi da 90, 2 porta cassette da 15 caduno con 60 dischi giochi-programmi e 30-35 cassette programmi - giochi. Tutto a lire 795.000 (trattabili). Scrivi a: Zito Pio, Via G. Matteotti, 13 71019 Vieste (FG) tel. 0884/76139

Club EPC vende ultimissime novità per C.64 a prezzi veramente incredibili spedizioni e abbonamenti in tutta Italia scrivere o telefonare a Mario Criscuolo Via Lucatelli, 35 cap. 00159 Roma tel. 06/4390265

Amiga 500/2000 vendo - scambio programmi nella zona Abbiategrasso - Vittuone - Corbetta tel. 9406648 ore 18.00/21.00 - Ciro Castoldi

Acquisto Amiga "500 " monitor 1084 + stampante MPS 1230 indirizzo: Via Pascoli, 20 71010 Peschici (FG) tel. 0884/94991 - Carlo Losito

Vendo cartuccia C.16 + introduzione al basic con 2 cassette a L. 60.000 tel. 7263120 Pianura (NA) ore seriali

Cerco disperatamente qualsiasi utility o gioco per C.64, scrivi a: Giovanni Bais Via Filangieri, 72 Vico Equense 80069 (NA)

Cerco mouse per Amiga 500, max 40.000, in buone condizioni. Telefonare ore pasti al 427889, Carlo

Cerco informazioni su goes/128 chi ne sa qualcosa scriva a Gelvarsi Giuseppe Via Riuscello 21/A 01017 Tuscania (VI) P.S. anche manuale

Possiedo modem 6499 - cerco utenti per collegamenti, preferibilmente stessa mia città - Carlo Cuzzola - Via P. Andiloro 13/A - tel. 09651/96660 - Reggio Calabria

Vendo - Commodore Plus 4 usato pochissimo + alimentatore + manuali + libro L. 250.000 trattabili - Scovenna Carlo - Via Emilia, 180 27043 - Broni (PV)

Vendo cassette per C.64 giochi vari edicola prezzo interessantissimo Massimo tel. 035-237002 Bergamo

Scambio software per Amiga nella provincia di Latina - scrivete a: Giannetti Paolo - Viale della Vittoria, 8 - 04019 Terracina (LT)

Cerco modem per C.64 e C.128 in buono stato completo di istruzioni - telefonare a Danilo Larizza Via Stadio a Valle, 24 Reggio Calabria tel. 0965/58434 (ore pasti)

Vendo - scambio - cerco programmi per C64 C128 inviare o richiedere liste a: Panzeri Nicola, tel. 039/590281 Via Belvedere, 12 22055 Merate (CO)

Compro, vendo, scambio software - consigli e consulenze per C64 scrivere a Pirozzi Enrico Via Pignotto, 19 Controguerra (TE) tel. 0861/89383

Vendo, eventualmente scambio, programmi di qualsiasi tipo per C64, su cassetta - Contatterei inoltre personalmente utenti della mia provincia, anche possedenti l'Amiga - Antonio Infanti Via Giorgione - Solighetto (TV) - tel. 0438/82020

Cerco disperatamente il turbo tape per il 128 max L. 3000 tel. 804646/095. Chiedere di Salvo Acicatena

Nuovo gruppo "realtime import " cerca utenti Amiga per scambio programmi e informazioni. Castiglioni Luca - Via Padre Ceriani, 19 20011 Corbetta (MI) - tel. 02/9778142

Vendo, scambio programmi per Amiga. Giorgio S. Vito Alt. (PN) tel. 0434/827558

C64 cartuccia simons basic originale vendo L. 20.000 Osvaldo Carlon Cann. 4588 - 30131 Venezia tel. 5283146 (cena)

Compro "Kit " pulizia testine per registratore a non più di 10.000 lire tel. 6224442

Vendo C64 new drive 1541 stampante 803 + 50 floppy, manuali, penna ottica, + speddos plus + adattatore telematico + mobile console (tutto 1.200.000) tel. 0142/54044 Ascoli Roberto

Cerco disperatamente il gioco "super mario bross" su cassetta per C64 pago bene. Tordi Fabio Via pineta, 3 27050 Brallo (PV)

C'è anche il tech club 64 Amiga non solo programmi anche recensioni e notizie - scrivere o telefonare a: Carrarini Andrea 06/5587998

Cerco utenti amiga per scambio software no lucro Alessandro Santel Via Ionie,1 32020 Rivamonte (BL) tel. 0437/65131

Cerco drive 1541 o 1571 per C64 a prezzo conveniente. Telefonare ore pasti a Lorenzo: 0464/71656

Vendo software per C64/128 scrivere a: Dino Calvisi Via dell'aia,1 67020 Petegona di Barisciano (AQ)

Scambio software CBM 64/128 solo su disk Zanolo Massimo Via Castello, 3 13030 Viterbo (VC) tel. 01663/450617

Volete gioco da sballo? Telefonate allo 0521/33536, oppure allo 0521/57407 la F.F Company vi soddisferà!!! (chiamare dalle 18,00 alla 19,00)

Cerco utenti modem per collegamenti & scambi Cava Alfredo tel. 0984/33570 - Cosenza

Cerco possessore Amiga 500 per scambio prog. e listati od idee. Contattare Schiavi Antonio Via Tombola, 19 Cera (VR) cap. 37053

Vendo, causa passaggio Amiga, drive compatibile C64 model 531 a L. 200.000 1 anno di vita - telfonare a: Nalli Gaetano V.G. Rotondi, 6 Milano tel. 02/4391879

Cedo corso video di autoistruzione basic per C64 20 lezioni su disco - Venturi Maurizio - Via Verdi,38 Gambettola (FO) tel. 0547/58837

Cerco disperatamente floppy disc C64 + dischetti e Joystik o mouse C64 Gian Maria tel. 0522/886239 ore pasti

Vendo modem per C64-128 a L. 50.000 mai adoperato. Rivolgersi al 0471/911262 - Rossi Marco

Cerco drive 1541 per C64 tel. 02/90753041 Fabrizio

DA VIDEO A STAMPANTE

Come trasferire, sulla stampante Commodore Mps 1200, i caratteri che compaiono nei vostri videogame preferiti

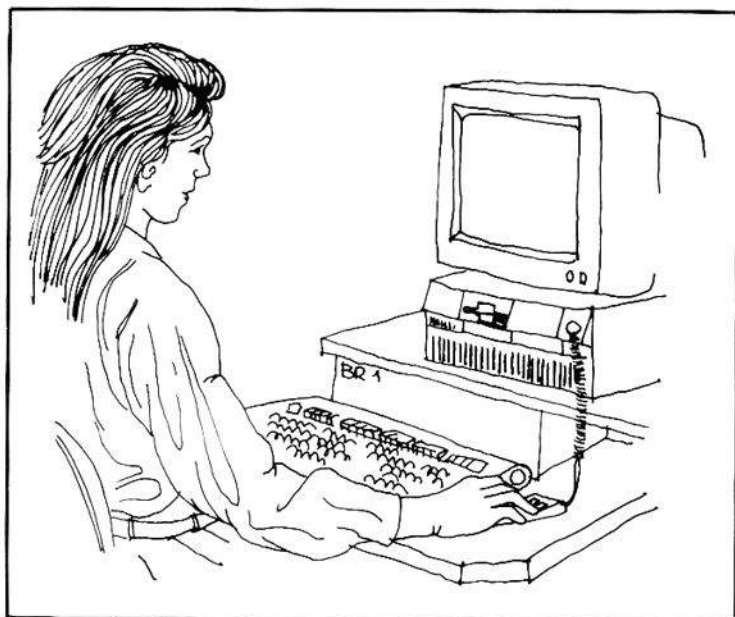
di **Bruno Cattivelli**

Il breve programma di queste pagine permette di stampare testi utilizzando font di caratteri programmati, tipici dei videogames, che di solito risiedodno nella memoria del C/64 anziché nella memoria della stampante.

Quest'ultima, infatti, non è completamente modificabile dall'utente il quale può agire solo sulle definizioni dei codici ASCII compresi fra 32 e 63, come descritto a pagina 8-5 del manuale; inoltre tale manovra si presenta piuttosto lunga e complessa.

Per utilizzare il listato è necessaria l'utilizzo di una stampante Commodore MPS 1200 (o una Epson FX) di cui viene sfruttata l'opzione "Epson Dots Graphics" (capitolo 6 del manuale della stampante) anziché il semplice modo grafico (capitolo 7). Ciò permette di inviare byte agli 8 aghi della testina che può così agevolmente stampare la matrice dei caratteri definiti nel C64, la quale, come tutti i lettori sanno, è formata da 8 x 8 punti. Tale scelta, quindi, comporta una maggiore semplicità di programmazione ed una maggiore velocità d'esecuzione.

SEQUENZE DI CODICI

L'opzione Epson dot graphic richiede unasequenza di escape (codice 27) di questo tipo:
chr$ (27) + "*" + chr$ (D) + chr$ (hbyte) + chr$ (lbyte)

in cui "D" determina la densità di stampa, e hbyte - lbyte esprimono la quantità di codici che la stampante deve interpretare come byte grafici invece che come normali codici ASCII (vedi capitolo 6 del manuale originale).

Il programma funziona come una macchina da scrivere elettronica su cui l'utente digita una riga di testo alla volta; la linea di input appare all'inizio dello schermo video e si possono apportare eventuali correzioni. La stampa vera e propria su carta avverrà premendo il tasto Return.

Il font di caratteri utilizzato dal programma deve essere situato in memoria dalla locazione 12288 alla locazione 16384 ($3000 - $4000); tale zona, del resto, è consigliata dalla "Guida di riferimento per il programmatore" del Commodore 64 per la definizione di caratteri da parte dell'utente. Si presume che, vista la mole di programmi pubblicati anche da CCC, molti lettori dispongano di fonts di caratteri alternativi e/o di utility atte a crearli. Per la creazione di caratteri si rimanda al capitolo 3 della suddetta guida.

La schermata iniziale illustra i valori di densità e di interlinea con i relativi codici; la prima richiesta del programma riguarda il nome di un file-prg su disco che contenga un font di caratteri autodefiniti.

Se l'utente non immette il nome del file, il programma copia il set standard del Commodore dalla rom alla ram ($3000 - $4000). In questo caso, quindi, la stampante utilizzerà il tipico carattere Commodore. Da notare che il carattere prescelto verrà utilizzato anche per la visualizzazione del testo sullo schermo (poke 53272, 31), e sarà in modo minuscolo, salvo modifica alla routine in LM. Da notare la possibilità di stampare anche il set reverse.

La seconda richiesta riguarda la densità di stampa che, nel nostro caso, si traduce in una maggiore o minore compressione dei caratteri. Una compressione accentuata migliora spesso l'estetica del carattere oltre a permettere un maggior numero di caratteri per riga, fino a 240 contro i 160 del compresso-elite della MPS-1200!

Il numero massimo di caratteri per riga corrisponde, curiosamente, al numero di dots per pollice (inch) in quanto la riga completa (senza margini) è lunga proprio 8 pollici e la matrice carattere è 8 x 8 dots. E' ovvio che la compressione di cui sopra non riguarderà i caratteri visualizzati sul monitor, ma soltanto quelli stampati su carta.

La terza richiesta riguarda l'ampiezza del "Line - spacing" (capitolo 4) ovvero dello spazio vuoto da lasciare tra una riga e l'altra. Si badi bene, quindi, che il valore 2, ad esempio, non significa che verranno lasciate due linee vuote, ma che verrà selezionata una ampiezza di 6 righe per pollice (ampiezza standard). Il valore 1, ad esempio, produrrà 10 righe per pollice che equivale ad annullare lo spazio tra una riga e l'altra permettendo di stampare disegni-intestazioni utilizzando i simboli grafici del set Commodore.

A questo punto il programma pulisce lo schermo: siamo in fase di editing. Premendo F1 torniamo alla schermata iniziale. Premere Return senza immettere testo, equivale a saltare una riga (come su qualsiasi macchina per scrivere). Nel digitare il testo dobbiamo tenere ben presente che, al momento di dare Return, il programma "legge", nelle locazioni 211 e 214, la posizione del cursore e assume automaticamente che essa sia la FINE della riga di input: in base ad essa calcola anche il numero di caratteri da trasformare in dots, passando questi valori alla routine in LM.

E' quindi fondamentale che la riga di input sia posta all'inizio dello schermo e che al momento di premere Return il cursore sia posizionato dopo l'ultimo carattere da stampare. Bisogna inoltre fare attenzione a non superare il numero massimo di caratteri per riga, perchè i caratteri in eccesso verranno ignorati e persi.

Il programma è qui presentato nella sua veste più semplice per due motivi:

1) per permettere una rapida e corretta digitazione del listato.

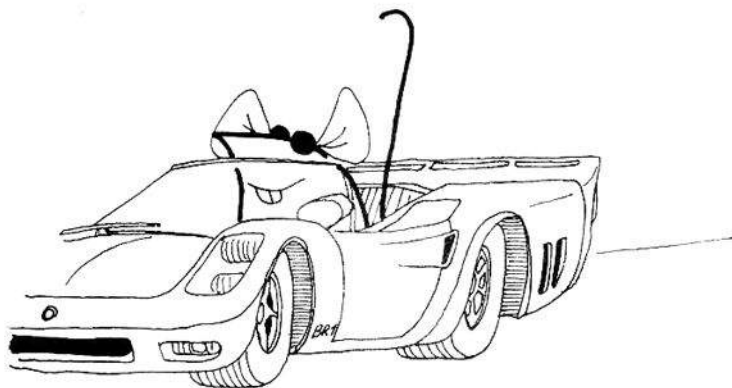
2) per stimolare il lettore ad apportare semplici modifiche per adattarlo agli altri vari usi che se ne possono fare.

Ecco alcune modifiche possibili:

- far stampare stringhe contenute in linee data, o in variabili, anzichè digitate in modo diretto
- far stampare, caricando una riga alla volta, un file - testo generato con easy-script (il quale, tra l'altro, permette di editare testi su 240 colonne) o con altri word - processors
- modificare, da Basic, la routine LM in modo da creare uno switch tra lower - case e upper - case (per l'utilizzo dei simboli grafici)
- richiedere, ed impostare, i margini del testo con le appropriate sequenze di escape richieste dalla stampante; ricordiamo, a tal proposito, che il carattere di escape, chr$(27), è presente nel listato come parentesi - quadra - reverse, ottenibile in modo "virgolette" premendo contemporaneamente CTRL e doppio punto (:).
- effettuare un controllo dell'input che non consenta di superare il numero massimo di caratteri per riga (selezionato in base alla densità scelta).
- solo per i più bravi: la routine in linguaggio macchina è strutturata in modo tale da essere facilmente manipolata.

E' possibile, ad esempio, ottenere rapidissime hard copy della memoria grafica del C64 col vantaggio (rispetto alle solite routine) di comprimere a piacimento il disegno ottenendo una migliore risoluzione.

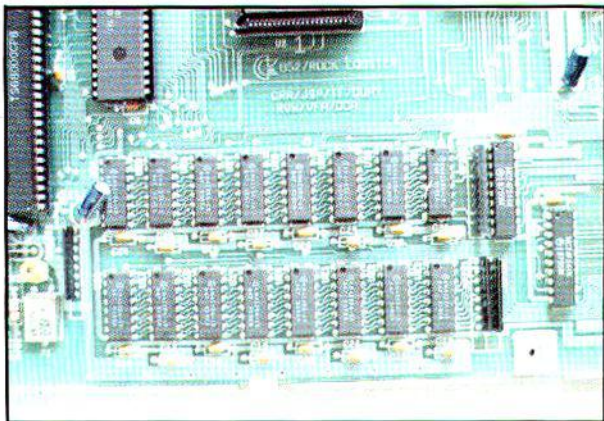
```
1 :rem font printer per c/64 e mps 1200
2 :
5 poke51,0:poke52,48:poke55,0:poke56,48
10 gosub400:open1,4,7:print#1,"@":n$=""
20 gosub200:input"Load font Carattere ";n$:ifn$<>""thenclose1:k=1:loadn$,8,1
25 poke53272,31:ifk=0thensys49266
30 input"Densita' di stampa    6";d
40 input"Interlinea (0-1-2)    2";a$
50 print#1,""a$;
100 print"";
120 poke204,0
130 geta$:ifa$=""then130
140 ifa$=chr$(13)then170
150 ifa$=""then20
160 wait207,1:poke204,1
165 printa$;:goto120
170 c=peek(214)*40+peek(211):q=c*8:n2%=q/256:n1=q-256*n2%:poke146,c
180 print#1,"*"chr$(d)chr$(n1)chr$(n2%);
190 sys49217:print#1,chr$(13);:goto100
200 printchr$(14)" C64 AutoFont Printer  *  S. Cattivelli ";
205 print"Densita' ottenibili:"
210 print"(0)  standard         60 bit/inch"
220 print"(1)  compresso       120"
225 print"(2)  compr.hi speed  120"
230 print"(3)  bicompresso     240"
240 print"(4)  largo            80"
250 print"(5)  largo+           72"
260 print"(6)  normalizzato     90"
270 print"(7)  compresso+      144"
280 print"(0)  interlinea    8 lines/inch"
290 print"(1)     ''         10 (no gap)"
300 print"(2)     ''          6"
310 return
400 reada,b:fori=atoa+b:reada:pokei,a:next:return
410 data 49152,158
500 data 32,48,192,169,0,168,170,224
501 data 8,240,36,192,8,240,18,177,251
502 data 61,60,3,240,8,165,2,24,121,60
503 data 3,133,2,200,208,234,160,0,165
504 data 2,32,210,255,169,0,133,2,232
505 data 208,216,96
506 data 169,1,162,8,157,59,3,10,202
507 data 208,249,162,1,32,201,255,96
508 data 32,48,192,162,0,160,0,134,253
509 data 228,146,240,35,189,0,4,133,251
510 data 169,0,133,252,162,3,24,6,251
511 data 38,252,24,202,208,248,165,252
512 data 24,105,56,133,252,32,3,192,166
513 data 253,232,208,213,96
514 data 120,169,51,133,1,169,0,133,251
515 data 133,253,169,48,133,252,169,208
516 data 133,254,162,0,160,0,177,253
517 data 145,251,200,208,249,232,230
518 data 252,230,254,224,16,208,238,169
519 data 55,133,1,88,96
520 end

ready.
```

```
DISASSEMBLATO COMMENTATO
AUTOFONT PRINTER - 64


legenda:

$92       (146)     :contiene la lunghezza della stringa
$fb-$fc (251-252):indirizzo della matrice del carattere da inviare
$02       (2)       :byte corrente in fase di calcolo


                    :ROUTINE DI INVERSIONE MATRICE CARATTERE ED INVIO
                    :DEGLI 8 BYTE RELATIVI ALLA STAMPANTE
                    :
C000 JSR $C030      :eventuale accesso per altri usi
C003 LDA #$00       :azzera accumulatore
     TAY            :azzera y
     TAX            :azzera x
C007 CPX #$08       :comincia la routine e controlla
     BEQ $C02F      :il ciclo esterno se x=8 finisce
C00B CPY #$08       :controlla ciclo interno se y=8
     BEQ $C021      :allora azzera x ed invia un byte
     LDA ($FB),Y    :ciclo interno
     AND $033C,X    :di inversione
     BEQ $C01E      :della matrice
     LDA $02        :del carattere
     CLC            :in $02 vi e' il
     ADC $033D,Y    :byte corrente
     STA $02        :in fase di calcolo
C01E INY            :se il ciclo interno
     BNE $C00B      :e' finito
C021 LDY #$00       :azzera il ciclo interno
     LDA $02        :carica il byte calcolato
     JSR $FFD2      :invia il codice alla stampante
     LDA #$00       :azzera il byte di calcolo
     STA $02        :posto in $02
     INX            :incrementa il ciclo esterno
     BNE $C007      :se e' finito
C02F RTS            :torna alla chiamata
                    :
                    :ROUTINE DI INIT VETTORE UTILITY E APERTURA CANALE
                    :
C030 LDA #$01       :crea in
     LDX #$08       :$033b - $0343
C034 STA $033B,X    :un vettore
     ASL            :esponenziale inverso
     DEX            :(128.64.32.16.8.4.2.1)
     BNE $C034      :che serve per l'inversione
     LDX #$01       :e apre il file logico #1
```

```
      JSR $FFC9     ;in output
C040  RTS           ;ritorna
                    ;
                    ;ROUTINE DI PRELIEVO DEI CODICI CARATTERE DALLA MEMORIA
                    ;DI SCHERMO E PASSAGGIO DELLA RELATIVA MATRICE A $C003
                    ;
C041  JSR $C030     ;entry point e init
      LDX #$00      ;azzera X
C046  LDY #$00      ;azzera Y
      STX $FD       ;imposta un contatore
      CPX $92       ;inizio ciclo, cfr X con lunghezza stringa
      BEQ $C071     ;se fine stringa allora end
      LDA $0400,X   ;prende codice schermo del carattere
      STA $FB       ;lo pone come byte basso in $fb
      LDA #$00      ;azzera il
      STA $FC       ;BYTE ALTO
      LDX #$03      ;calcola
      CLC           ;e pone in
C05A  ASL $FB       ;$fb
      ROL $FC       ;$fc
      CLC           ;byte basso
      DEX           ;e byte alto
      BNE $C05A     ;della
      LDA $FC       ;matrice
      CLC           ;corrispondente
      ADC #$38      ;al
      STA $FC       ;carattere in corso di stampa
      JSR $C003     ;alla routine di inversione matrice e stampa
      LDX $FD       ;aggiorna
      INX           ;il contatore
      BNE $C046     ;se contatore < 255 ricomincia
C071  RTS           ;torna al basic
                    ;
                    ;ROUTINE DI COPIATURA DEL SET CARATTERI DALLA ROM ALLA RAM
                    ;
C072  SEI           ;elimina l'interrupt
      LDA #$33      ;apre l'accesso alla rom
      STA $01       ;carattere agendo su $01
      LDA #$00      ;azzera i byte bassi
      STA $FB       ;della ram e
      STA $FD       ;della rom
      LDA #$30      ;imposta i byte alti
      STA $FC       ;della ram
      LDA #$D0      ;e della
      STA $FE       ;rom
      LDX #$00      ;azzera contatore ciclo esterno
C087  LDY #$00      ;azzera contatore ciclo interno
C089  LDA ($FD),Y   ;copia carattere dalla rom
      STA ($FB),Y   ;nella ram
      INY           ;incrementa contatore
      BNE $C089     ;controllo ciclo interno
      INX           ;incrementa contatore esterno
      INC $FC       ;inc byte alto ram
      INC $FE       ;inc byte alto rom
      CPX #$10      ;controllo ed eventuale
      BNE $C087     ;uscita dal ciclo esterno
      LDA #$37      ;ripristino dei valori
      STA $01       ;di default del sistema
      CLI           ;reimposta interrupt
      RTS           ;torna al basic
```

RAM RISPARMIATA, RAM GUADAGNATA

Come usare set di caratteri alternativi senza sprecare inutilmente memoria Ram

di **Walter Adriani**

Nel realizzare programmi Basic in bassa risoluzione, può capitare di dover modificare il set di caratteri per crearne di nuovi che rispondano meglio alle nostre esigenze.

Il modo più semplice per risolvere il problema è di trasferire il set originale Commodore in una zona RAM, in modo da modificarlo a volontà. Se, però, non si prendono determinati accorgimenti, la memoria disponibile viene notevolmente ridotta, e basta qualche matrice pluridimensionale a generare il noioso ?OUT OF MEMORY ERROR.

Ciò avviene perchè il VIC-II (circuito integrato grafico del C/64) può controllare uno solo dei 4 banchi di 16K ciascuno in cui si suddividono i 64K totali. Ne consegue che se la memoria di schermo si trova nel primo banco, qui devono trovarsi anche le informazioni relative ai caratteri.

LA TEORIA

Copiando il set di caratteri nella RAM del primo banco, il punto più alto da cui può partire è la locazione 12288 (14336 se si copia un solo insieme, minuscole o maiuscole). E' quindi necessario abbassare il top di memoria per evitare sovrapposizioni tra le variabili o tra il programma ed i dati dei caratteri. La memoria RAM disponibile per i programmi diventa perciò di soli 12K (dalla locazione 2048 alla 14336), vale a dire appena 1/3 dei 40K normalmente accessibili al Basic.

Per ovviare a questo problema, dovremmo posizionare il set di caratteri in una zona RAM che non sia quella accessibile al Basic, cioè in quella che va da $C000 a $CFFF. Questa si trova, però, nel banco 3, mentre lo schermo è nel banco 0. Come risolvere il dilemma? Semplice: spostando anche la memoria di schermo!

Ecco, pertanto, come ripartire i 4K dell'area RAM suddetta (da $C0000 a $CFFFF): 1K servirà per la memoria di schermo, 2K per allocare un insieme di caratteri; rimane addirittura libero 1K per eventuali routine L.M. (oltre quella del presente articolo). Altri 1024 byte sono liberi nell'ex-area di schermo standard (da 1024 a 2023). Il nuovo schermo, infatti, partirà da 49152 e terminerà a 50151, i caratteri (modificabili) si troveranno tra 51200 e 53247.

La routine di queste pagine esegue tutto il lavoro necessario per le dovute modifiche. Dapprima copia l'insieme maiuscole / grafici a partire da $C800, poi imposta il VIC-II al banco 3 in modo che possa prelevare correttamente le informazioni relative al video (a partire da $C000) e quelle relative ai caratteri (a partire da $C800).

Per spostare il banco di memoria si usano i 2 bit bassi della locazione $DD00. Il loro valore di default, %11 (banco 0), viene resettato a %00 (banco 3). Per far ciò dobbiamo prima impostare ad OUTPUT (stato logico 1) i corrispondenti bit del registro direzione dati allocato a $DD02.

Per spostare le memorie video e carattere si usano, rispettivamente, i bit 7 - 4 e 3 - 1 del registro $D018. I primi selezionano il blocco di 1K che, all'interno del banco, deve fornire l'informazione video (valore di default: %0001): il valore da immettere è %0000 (primo blocco di 1K del banco 3, 49152).

I secondi selezionano il blocco di 2K, sempre all'interno del banco, che deve fornire l'informazione carattere (valori di default: %010 per le maiuscole e %011 per le minuscole): nel nostro caso verrà impostato il valore %001 (secondo blocco di 2K

DA UN BANCO ALL'ALTRO

Il programma pubblicato è nato per evitare spreco di RAM quando si vogliono modificare i caratteri, ma la tecnica per spostare lo schermo è valida ed utilizzabile per qualsiasi altro tipo di applicazione.

Si potrebbe, ad esempio, creare un secondo schermo di riserva, richiamabile con un tasto, consultabile mediante lo split di RASTER oppure la tecnica delle WINDOWS, a patto di realizzare la routine L.M. appropriata.

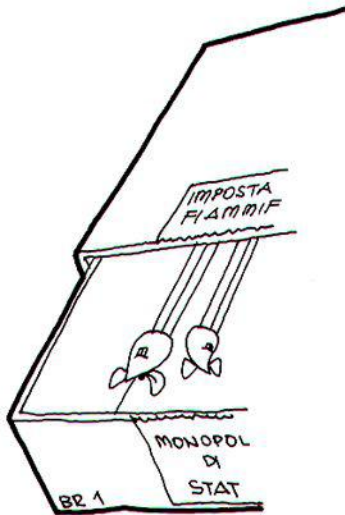
Nel caso si voglia posizionare lo schermo in parti della memoria diverse da $C000, la tabella qui riportata fornisce i valori da registrare nelle locazioni $DD00 e $D018.

Non è comunque consigliabile creare più di tre memorie schermo, perchè esse occupano pur sempre 1K di memoria ciascuna (+ 2K per i caratteri, se si trovano in un diversi banchi video).

E' bene sottolineare che il valore da registrare in $D018 è composto dai 4 bit (alti) relativi alla memoria di schermo più i 3 bit (bassi) relativi alla memoria carattere. Il bit 0 non viene utilizzato.

IMPOSTA FIAMMIF

MONOPOL DI STAT

BR 1

del banco 3, 49152 + 2048 = 51200). Sommando i due valori si ottiene %0000001x, che il programma registra in $D018.

Riguardo allo spostamento dell'area di schermo, dobbiamo fare un'importante precisazione. Costringere il VIC-II a prelevare l'informazione video a partire da una locazione diversa da $0400 non significa aver spostato la memoria di schermo: infatti, pur vedendo un'altra sezione di memoria, ciò che digitiamo, ed i messaggi che il computer visualizza, continuano a trovarsi nello schermo di default. C'è infatti una locazione, estranea al VIC-II, che indica al computer in che pagina si trova la memoria di schermo, cioè dove devono essere pokati i codici video: è la locazione 648 ($0288), che normalmente contiene il valore $04.

Se vogliamo spostare integralmente l'area di schermo, dobbiamo alterare tale valore (che altro non è se non il byte alto di inizio della memoria di schermo): il valore da immettere è quindi $C0.

Un bug del C/64 impedisce ai tasti Run/Stop e Restore di ripristinare questa locazione (ripristinano, invece, i registri del VIC-II). Perciò, per tornare alle condizioni normali, oltre a Run/Stop e Restore si deve digitare alla cieca POKE 648, 4 (il messaggio non viene visualizzato) e premere il tasto Return: sullo schermo apparentemente morto tornerà il solito READY.

PER USARE LA ROUTINE

Per attivare la routine, allocata a partire da 50176 ($C400) basterà una semplice SYS 50176. Per modificare i caratteri si può agire da Basic, come indicato nel riquadro pubblicato. Chi ne è capace, può sfruttare la velocità del L.M.

Un'ultima avvertenza: spostando la pagina dello schermo (locazione 648), si spostano anche i puntatori degli sprite: invece che tra 2040 e 2047, si troveranno tra 50168 e 50175, ed i loro dati dovranno trovarsi all'interno del banco 3, nel Kbyte di RAM rimasto libero.

UN NUOVO CARATTERE

La routine pubblicata serve a spostare la memoria carattere su RAM, affinchè sia possibile modificare il set di caratteri. Le poche note che seguono spiegano in maniera esauriente come si deve procedere per modificare ogni singolo carattere a nostro piacimento.

Dopo aver mandato in esecuzione la routine, a partire da 51200 ($C800) si troveranno le configurazioni di bit che formano i caratteri. Ciascuno di essi è formato da una griglia 8 x 8 in cui ogni quadratino corrisponde ad un bit, "acceso" o "spento", ed ogni riga ad un byte. Un carattere è perciò definito da 8 byte.

I codici POKE (attenzione: POKE e non ASCII) indicano la posizione che il carattere occupa all'interno dell'insieme. E' possibile calcolare la locazione di partenza di ogni carattere con la formula...

L = 51200 + (codice POKE) x 8

...e modificarlo con il LOOP...

```
FOR I=L TO L+7: READ A: POKE I, A: NEXT
DATA ... (8 valori)
```

Gli 8 valori da inserire nelle linee DATA sono numeri interi che corrispondono alla configurazione di bit di ciascuno degli 8 byte.

Il procedimento per calcolare i valori è analogo a quello per ricavare i DATA degli sprite, e non dovrebbe presentare difficoltà.

E' sufficiente disegnare (su un foglio di carta) il carattere desiderato all'interno della griglia 8 x 8, "annerendo" i bit che si intendono vedere "accesi" sullo schermo. Si dovranno poi calcolare i valori decimali corrispondenti a ciascuna riga, tenendo presente che esse sono byte in cui i bit "accesi" valgono 1 e quelli "spenti" 0.

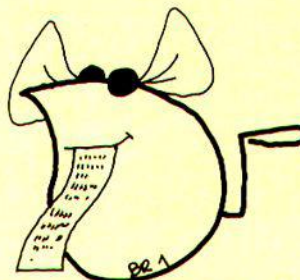
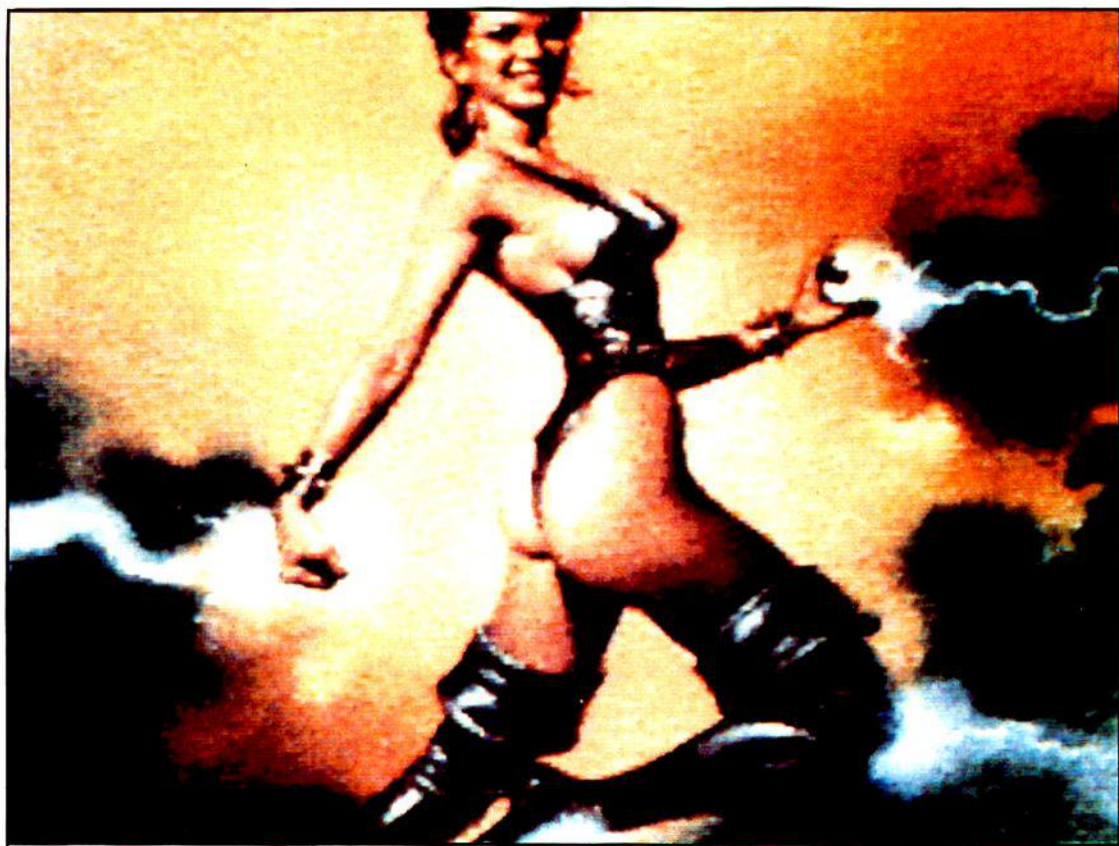
La conversione da binario a decimale non dovrebbe rappresentare un problema per nessuno: in ogni caso nel manuale del C/64 si trova un programma adatto allo scopo.

Se, ad esempio, in una riga sono "accesi" i bit 7, 4, 3 ed 1 (i bit si numerano da 7 a 0 partendo da sinistra), il byte si presenterà così: %10011010 ,ed il valore da immettere nei DATA sarà 154.

```
**********************
*                    *
* Disassemblato routine *
*   'SPOSTA SCHERMO'   *
*   By  Walter Adriani *
*                    *
**********************

Allocata da     : $C400

---------------------------------------
c400 sei             Disattiva interrupt
c401 lda $d011       Disattiva lo
c404 and #$ef         schermo
c406 sta $d011
c409 lda $01         Seleziona ROM genera-
c40b and #$fb         trice caratteri
c40d sta $01
c40f ldy #$00        Indice Y = 0
c411 lda $d000,y     --------------------
c414 sta $c800,y     LOOP che copia
c417 lda $d100,y      il 'set' caratteri
c41a sta $c900,y     maiuscoli/grafici
c41d lda $d200,y     nelle locazioni
c420 sta $ca00,y     $C800-$CFFF
c423 lda $d300,y
c426 sta $cb00,y
c429 lda $d400,y     Per copiare
c42c sta $cc00,y     il 'set' caratteri
c42f lda $d500,y     minuscoli/maiuscoli
c432 sta $cd00,y     sostituire le loca-
c435 lda $d600,y     zioni $D800-$DFFF
c438 sta $ce00,y     alle $D000-$D7FF
c43b lda $d700,y
c43e sta $cf00,y     --------------------
c441 iny             Incrementa indice Y
c442 cpy #$00        Indice Y = 0 ?
c444 bne $c411       No: continua il LOOP
c446 lda $01         Si: ripristina I/O
c448 ora #$04         (Fine LOOP)
c44a sta $01
c44c lda $dd02       Bits 1-0 loc. $dd02
c44f ora #$03         (reg. direz. dati)
c451 sta $dd02        settati ad OUTPUT
c454 lda $dd00
c457 and #$fc        Seleziona il banco 3
c459 sta $dd00
c45c lda $d018       Imposta nuove memorie
c45f and #$01         video e carattere
c461 ora #$02         nel VIC-II
c463 sta $d018
c466 lda #$c0        Imposta pagina video
c468 sta $0288        a $C0
c46b lda #$93        Invia un CLR
c46d jsr $ffd2        allo schermo
c470 lda $d011
c473 ora #$10        Riattiva lo schermo
c475 sta $d011
c478 lda #$08        Disattiva 'switch'
c47a jsr $ffd2        SHIFT/Commodore
c47d lda #$0e        Seleziona 'set'
c47f jsr $ffd2        copiato su RAM
c482 cli             Riattiva interrupt
c483 rts             Torna al BASIC
---------------------------------------
```

```
100 REM ***************************
105 REM *       (SOLO PER C/64)       *
110 REM * ROUTINE 'SPOSTA SCHERMO' *
115 REM *     BY  WALTER ADRIANI      *
120 REM *      BRACCIANO (ROMA)       *
125 REM *                             *
130 REM ***************************
140 :
150 FORR=0TO131:READA:POKE50176+R,A
155 C=C+A:NEXT
160 IFC<>15488THENPRINT"ERRORE":END
170 SYS 50176
180 POKE53280,0:POKE53281,0
190 PRINT"[CLR] SCHERMO POSIZIONATO A $C000"
200 END
1000 DATA 120,173,017,208,041,239,141
1001 DATA 017,208,165,001,041,251,133
1002 DATA 001,160,000,185,000,208,153
1003 DATA 000,200,185,000,209,153,000
1004 DATA 201,185,000,210,153,000,202
1005 DATA 185,000,211,153,000,203,185
1006 DATA 000,212,153,000,204,185,000
1007 DATA 213,153,000,205,185,000,214
1008 DATA 153,000,206,185,000,215,153
1009 DATA 000,207,200,192,000,208,203
1010 DATA 165,001,009,004,133,001,173
1011 DATA 002,221,009,003,141,002,221
1012 DATA 173,000,221,041,252,141,000
1013 DATA 221,173,024,208,041,001,009
1014 DATA 002,141,024,208,169,192,141
1015 DATA 136,002,169,147,032,210,255
1016 DATA 173,017,208,009,016,141,017
1017 DATA 208,169,008,032,210,255,169
1018 DATA 014,032,210,255,088,096
1019 END
```

```
100 REM DEMO SPOSTA SCHERMO
110 REM ACRICARE E LANCIARE ROUTINE LM
120 PRINT CHR$(147)"ATTENTO...":PRINT
130 FOR I=1 TO 40: PRINT CHR$(64);:NEXT
140 PRINT
150 FOR I=1 TO 40:PRINT CHR$(65);:NEXT
160 PRINT:PRINT "ORA RIEMPO "CHR$(64)
170 GOSUB 250
180 FOR I=51200 TO 51200 + 7
190 POKE I,255: NEXT
200 PRINT:PRINT"ORA RIEMPO "CHR$(65)
210 GOSUB 250
220 FORI=51207 TO 51207 + 7
230 POKEI,170:NEXT
240 END
250 PRINT"PREMI UN TASTO"
260 GET A$: IF A$= "" THEN 260
270 RETURN
```

SHELL, OVVERO IL NUOVO CLI

Come utilizzare un nuovo comando disponibile con Workbench 1.3

di **Roberto Buzzi**

La nuova versione di Workbench (la 1.3) presenta notevoli vantaggi, sia per il trasferimento dati da/per il disco, sia per l'uscita su stampante. Entrambe le operazioni risultano notevolmente più veloci e la casa madre, addirittura, parla di velocità di stampa da due a TRENTA volte superiori rispetto ai driver della versione 1.2. Inoltre molti programmi inseriti sul dischetto sono stati modificati e potenziati ed altri nuovi sono stati aggiunti.

Purtroppo, però, per utilizzare al meglio le nuove potenzialità bisogna cambiare la ROM del Kickstart, sia esso il 500 o il 2000 (discorso a parte per l'AMIGA 1000, in quanto funziona con il Kickstart su disco). La cosa è però poco conveniente per il semplice motivo che la Commodore ha già annunciato (in occasione della presentazione della versione 1.3) che è già in fase di studio, e realizzazione, la versione 1.4, per cui....

Comunque la versione 1.3 presenta innovazioni molto utili.

Una di queste è SHELL e servirà, almeno in parte, a capire il funzionamento delle altre novità.

Digitando LIST compaiono, di solito, gli indicatori (flag) "r w e d"; questi, come è noto, significano, rispettivamente, "read", "write", "execute", "delete" e, a seconda se presenti o meno, indicano la possibilità che un file possa essere letto, scritto o modificato, eseguito e cancellato. Con Workbench 1.3 possono apparire altri quattro flag "h s p a" che significano "hidden" (nascosto), "script" (che identifica un file batch, ossia un file di comandi CLI), "pure" (che indica i comandi CLI che pos-

sono essere resi residenti), "archive" (archivio). Ovviamente i nuovi indicatori possono essere modificati usando il comando PROTECT, esattamente come gli altri indicatori che già conosciamo.

La novità più valida è, comunque, SHELL, una specie di evoluzione dello stesso CLI (Command Line Interface).

I COMANDI DI SHELL

Con SHELL è possibile ricorrere ai tasti "Freccia a destra" e "Freccia a sinistra" per spostarsi all'interno della linea di comando in modo da introdurre, conseguentemente, le eventuali modifiche senza essere obbligati a ribattere l'intero comando (come invece succedeva con la precedente versione).

Premendo gli stessi tasti, ma in unione con Shift, si manda il cursore, rispettivamente, all'inizio ed alla fine della linea.

Con Control "X" si cancella tutto ciò che è posizionato alla sinistra del cursore.

Control "U" inserisce spazi alla destra del cursore.

SHELL permette di utilizzare i comandi CLI in modo residente, cioè presenti in memoria senza doverli caricare ogni volta dal disco di sistema o dal RAM-disk.

Per fare ciò bisogna rendere residente il comando CLI utilizzando il comando RESIDENT < NOME comando >. Per utilizzare i comandi in questo modo occorre che l'indicatore del file sia settato su "P".

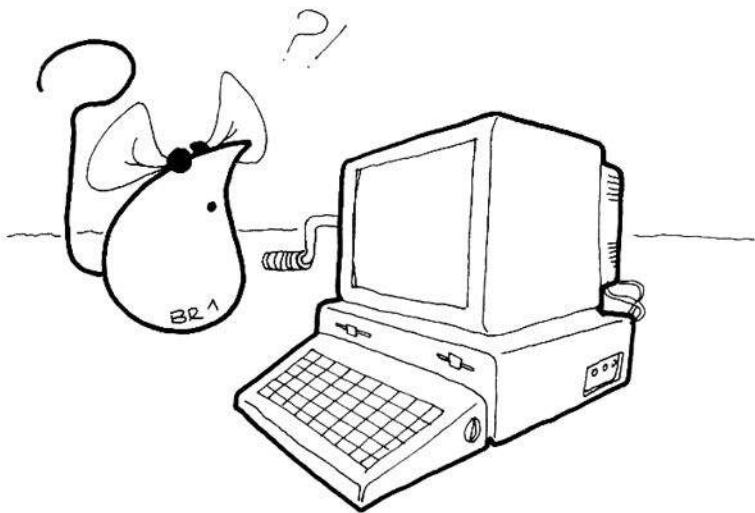
Altra possibilità data da SHELL è quella di eseguire un file di comandi (file batch) senza utilizzare il comando EXECUTE < nome file >, ma semplicemente indicando, sulla linea comandi, il nome del file da eseguire. Occorre che l'indicatore del file sia settato su "S".

Inoltre SHELL permette di memorizzare i comandi immessi volta per volta da tastiera in un buffer di memoria di ben 2 Kbyte. Questo permette di richiamarli a piacere, magari per eseguirli nuovamente, così come sono o modificandoli a piacimento. Per fare ciò si usano i tasti "Freccia in alto" e "Freccia in basso".

Con gli stessi tasti, premuti insieme a Shift, è possibile richiamare il comando che si avvicina a quello della linea corrente, o posizionato in fondo al buffer.

Continuando a descrivere i vantaggi di SHELL, c'è da dire che è possibile utilizzare gli "ALIAS". Se, per esempio, digitiamo sulla linea comandi (o tramite un file comandi eseguibile)...

ALIAS CIAO DIR

...otteniamo che ogni volta che verrà digitata la parola "CIAO" il sistema eseguirà il comando DIR (il comando "DIR" verrà comunque sempre riconosciuto).

Appena si attiva uno SHELL da icona o da CLI, con il comando NEWSHELL il sistema ricerca nel cassetto "S" se esiste un file chiamato SHELL-STARTUP e, in caso positivo, lo esegue: così è possibile definire una serie di "alias" e/o rendere residenti i comandi desiderati (per creare il file shell-startup vedi più avanti).

Per finire, il prompt (l'indicatore che segnala la linea comandi) indica il disco ed il cassetto attualmente in uso dando così la possibilità di conoscere in ogni momento in quale anfratto del disco ci si trova.

AUTOSHELL

Vediamo ora come fare per costruire un disco che parta immediatamente con SHELL.

Ovviamente bisogna per prima cosa procurarsi un Workbench versione 1.3, poi formattare un dischetto vergine (chiamandolo, per esempio, "AutoSHELL") per mezzo dei comandi FORMAT e, subito dopo, INSTALL.

Poi bisogna creare le directory C, L, Devs, e S usando il comando MAKEDIR. Copiare l'intera directory C, copiare nella directory L i file "NewCon-Handler" e "Shell-Seg" che si trovano nell'omonimo cassetto del disco WB 1.3, copiare nella directory Devs il file "Mountlist", nella directory S il file "CLI-Startup" ed infine creare la seguente Startup-Sequence da inserire nella directory S:

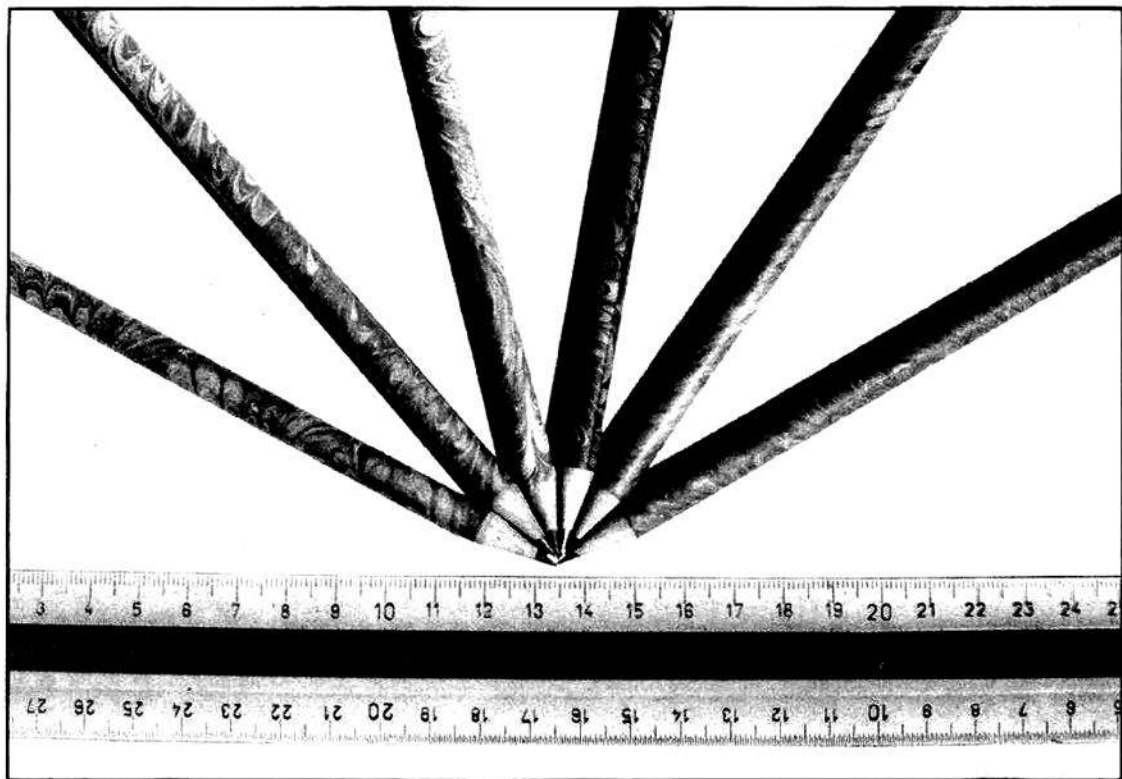
```
Resident CLI L:Shel-Seg System pure
Resident C:Execute pure
Mount Newcon:
NewCLI    Newcon:0/10/640/240/Auto
SHELL
EndCLI
```

Le prime due righe rendono residenti, rispettivamente, il file SHELL-Seg e il comando Execute, la terza carica Newcon-Handler, che sono i parametri per il sistema che si trovano nella MountList, la quarta crea una finestra CLI con 0 / 10 (angolo superiore sinistro) 640 / 240 (angolo inferiore destro) con nome "AutoSHELL", la quinta chiude la finestra AmigaDOS.

A chi non sapesse come fare per creare un file tipo la "Startup-Sequence", ricordiamo che bisogna digitare i seguenti comandi nella finestra CLI:

```
cd AutoSHELL:
ed s/startup-sequence
```

A questo punto il sistema apre una finestra su quella di CLI, in cui digitare quanto scritto sopra. Alla fine occorre premere il tasto ESC (azione che fa comparire un asterisco in basso a sinistra nella finestra di editor) e subito dopo il tasto "X". Il sistema provvede quindi a registrare sul disco AutoSHELL, nel cassetto S, il file chiamato Startup-Sequence che, oltretutto, è il file che inizializza il sistema.

Bisogna prestare molta attenzione, specialmente se si possiede solo il drive interno ad AMIGA, in quanto verranno richiesti diversi cambi di dischi. Inoltre consigliamo vivamente di proteggere contro la scrittura il disco originale WB 1.3 che dovrebbe, però, essere una copia di quello che avete trovato nella confezione: mai lavorare con gli originali!

HILINER, ATTO II

Spesso i programmi che pubblichiamo offrono uno spunto per realizzare listati più sofisticati; un nostro lettore, ad esempio...

di **Emilio Gelosi**

Il programma che pubblichiamo è la rielaborazione di un listato apparso sul numero di ottobre '88 di C.C.C.

Il suddetto utilizzava gli sprite per evidenziare la riga attualmente occupata dal cursore, in modo da facilitare il controllo e la rilettura di un qualunque listato, in particolare se contenente numerose righe di tipo Data.

La tecnica usata rappresentava il massimo della funzionalità, specie perché il programma doveva esser disattivato durante le operazioni di I/O a causa di un noto bug del S.O. del C/64, per non parlare della necessità di "occupare" molti sprite allo scopo di evidenziare la riga video.

La routine che riportiamo in queste pagine, invece, utilizza il controllo del raster, permettendo di visualizzare, nella riga in cui si trova il cursore, una linea gialla larga quanto l'intero schermo.

Le operazioni svolte per calcolare il punto di attacco del pennello elettronico sono in pratica analoghe a quelle del precedente programma.

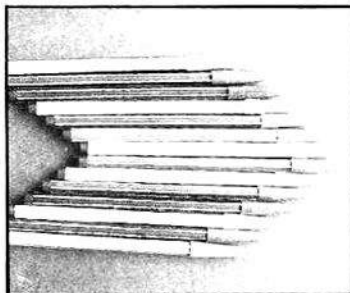
BREVI NOTE SUL PROGRAMMA

Il ritardo presente all'inizio della routine in interrupt serve per evitare alcuni lievi sfarfallii. Volendo maggior spazio in memoria è possibile sopprimerlo.

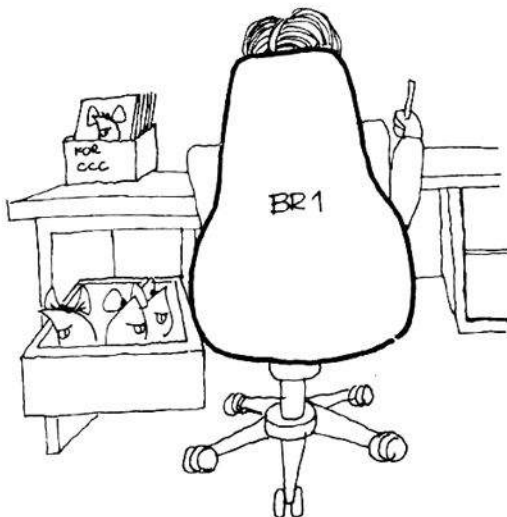
Una versione per il C/128, in modo 128, è sicuramente possibile. Bisogna solo tenere conto della locazione in cui è memorizzata la riga ove si trova il cursore ($EB al posto di $D6) e del fatto che, per uscire dall' interrupt, bisogna saltare a $FA65 invece che a $EA31.

IL LISTATO

Il programma in Basic contiene un mini-caricatore che consente di allocare in memoria i codici macchina a partire da 49152. Come al solito, in caso di errore un'opportuna segnalazione impedisce di attivare la Sys che provocherebbe disastri.

Il disassemblato commentato sarà apprezzato dagli smanettoni ai quali, in effetti, sono dedicate le presenti note.

```
9000 REM HILINER 64  BY EMILIO GELOSI
9010 FOR X=49152 TO 49226
9020 READ A:POKE X,A:CK=CK+A
9030 NEXTX
9040 IF CK=9007 THEN 10080
9050 PRINT "ERRORE NEI DATA":END
10000 DATA 120,169,127,141,013,220,045,017,208,141
10010 DATA 017,208,169,129,141,026,208,169,029,141
10020 DATA 020,003,169,192,141,021,003,088,096,173
10030 DATA 025,208,141,025,208,162,002,202,208,253
10040 DATA 169,007,141,032,208,141,033,208,162,103
10050 DATA 202,208,253,169,000,141,032,208,141,033
10060 DATA 208,165,214,010,010,010,105,049,141,018
10070 DATA 208,076,049,234,141
10080 PRINT"ROUTINE ALLOCATA PARTIRE DA 49152"
10090 SYS 49152:END
```

HILINER 2
DISASSEMBLATO COMMENTATO

```
*=$c000
vicreg=$d012
vicflg=$d019
vicmsk=$d01a
bordo=$d020
sfondo=$d021
 sei              ;disab.interrupts
 lda #$7f         ;setta
 sta $dc0d        ;il timer
 and $d011        ;azzera il
 sta $d011        ;msb di raster
 lda #$81         ;attiva le
 sta vicmsk       ;irq raster
 lda #<main       ;fa puntare
 sta $314         ;le irq
 lda #>main       ;a
 sta $315         ;main
 cli              ;riab.interrupts
 rts              ;ed esce
 ;------------    ;----------------;
main lda vicflg   ;riattiva le
 sta vicflg       ;raster interr.
 ldx #$02         ;ritardo
lp1 dex           ;per evitare'
 bne lp1          ;sfarfallii
 lda #$07         ;mette giallo
 sta bordo        ;in bordo
 sta sfondo       ;e sfondo
 ldx #$67         ;e dopo 10
lp2 dex           ;pixels
 bne lp2          ;fa tornare
 lda #$00         ;nero
 sta bordo        ;bordo
 sta sfondo       ;e sfondo
 lda $d6          ;calcola
 asl              ;la
 asl              ;nuova posizione
 asl              ;dell'irq
 adc #$31         ;di raster
 sta vicreg       ;ed esce
 jmp $ea31        ;dagli interr.
;-----------      ;--------------
;1440 .end
```

UN PROBLEMA BILINGUE

Esaminiamo da vicino, risolvendo un problema classico, alcune differenze tra Basic e Turbo Pascal

di **Alessandro de Simone**

Come i più affezionati lettori di C.C.C. già intuiscono, affronteremo lo studio proposto in modo diretto, cioè servendoci del sistema più rapido per apprendere un qualsiasi argomento.

Ci stiamo riferendo, ovviamente, alla procedura più semplice per seguire un ragionamento con la massima efficacia:

1- Scelta di un problema suscettibile di soluzione mediante calcolatore.

2- Stesura di un listato in Basic in grado di risolvere il problema.

3- Stesura di analogo programma in linguaggio alternativo (Turbo Pascal, nel nostro caso).

4- Esame delle somiglianze e differenze esistenti tra i due linguaggi.

5- Suggerimenti sul da farsi nel caso (inevitabile) di vari inconvenienti incontrati sviluppando il punto 3.

Diamo infatti per scontata la padronanza del linguaggio Basic (almeno per ciò che riguarda le più elementari istruzioni) come pure la capacità, da parte di chi ci vuol seguire, di intervenire immediatamente nel caso di segnalzzione di errori. Gli

SCHEDA TECNICA

Articolo didattico per il confronto tra due linguaggi
Hardware richiesto:
Basic: qualsiasi computer Commodore;
Turbo Pascal: C/128 + disk drive:
Oppure qualsiasi sistema computerizzato in grado di supportare Basic e T. Pascal
Sistema operativo richiesto:
C/128: CP/M:
Ms/Dos compatibili: Ms-Dos
Linguaggio richiesto: Turbo Pascal Borland
Consigliato ai lettori più evoluti

altri possono comunque proseguire nella lettura; in ogni caso non sarà tempo perso.

IL BREAKEVENPOINT

Un noto problema, basilare in un qualunque ragionamento "economico", è quello relativo alla determinazione del numero di pezzi minimo da produrre affinchè un'industria possa pareggiare spese e ricavi.

Spieghiamoci meglio. Supponiamo che un'industria acquisti un macchinario per un milione di lire. Grazie al suo impiego, la spesa relativa a ciascun oggetto prodotto risulta di lire 400 (materiale, energia ed altro, tutto compreso). Lo stesso oggetto viene poi posto in vendita a L. 600. Se, per assurdo, riuscissimo a produrre un solo oggetto (ad esempio perchè, subito dopo, i soliti ignoti rubano il macchinario e dobbiamo interrompere la produzione), questo ci verrebbe a costare L. 1.000.400 (L. 400 dell'oggetto + costo macchinario). Rivendendo l'oggetto a L. 600 (è impensabile rivenderlo a L. 1.000.600!) avremmo una perdita secca di L. 999.800 (L. 600 - 1.000.400).

Se, però, prima di farci rubare il macchinario, siamo riusciti a realizzare (e vendere) mille pezzi, risulterà:

Costo: 1.000.000 + 1000 * 400 = 1.400.000; ricavando L. 600 * 1000 = 600.000 la perdita sarà minore di prima (800.000 lire).

Quanti oggetti, allora (ecco il problema), occorrerà produrre affinchè il ricavo pareggi almeno le spese, nella peggiore delle ipotesi?

Indicando con CF il Costo Fisso del macchinario (o della procedura in senso generale), con CU il Costo per ciascuna Unità, con RU il Ricavo per Unità, risulta vero che (BE= Breakevenpoint)...

BE = CF / (RU - CU)
BE = 1.000.000 / (600 - 400)
BE = 1.000.000 / 200
BE = 5.000

...bisognerà produrre 5.000 pezzi. Il risultato indica anche, implicitamente, che una produzione superiore alle 5.000 unità consentirà un guadagno; al contrario, una produzione inferiore rappresenterà una perdita, più o meno accentuata, da determinare con altre, semplici relazioni matematiche sulle quali, tuttavia, non ci intratteniamo.

IL PROGRAMMA BASIC

Una delle possibili "traduzioni" del problema in linguaggio Basic è rappresentata dal listato di queste pagine.

Per semplificare le cose, i nomi delle quattro quantità sono quelli già visti (CF, CU, RU, BE). Alcuni "filtri" software, sui quali ci intratterremo tra breve, sono stati introdotti per generalizzare il problema.

I casi che possono presentarsi, infatti, sono numerosi, soprattutto considerando che l'utilizzatore del programma può essere distratto (o pignolo).

Il ricorso al programma può aver senso solo se c'è, in effetti, un costo fisso (CF). In caso contrario è sufficiente confrontare il costo (CU) ed il ricavo (RU); se il secondo è superiore al primo ha senso elaborare i dati; in caso contrario la situazione pro-

UNA SECONDA LINGUA

Al giorno d'oggi, scrivere un programma in un linguaggio, piuttosto che in un altro, è la stessa identica cosa, almeno in termini di funzionamento (e soddisfazioni personali). Un qualsiasi linguaggio, in altre parole, consente di portare a termine elaborazioni di qualunque tipo.

Se un wordprocessor, o un più modesto videogame, sia stato scritto in "C", in linguaggio macchina o in Pascal, all'utente finale non interessa minimamente: basta che funzioni.

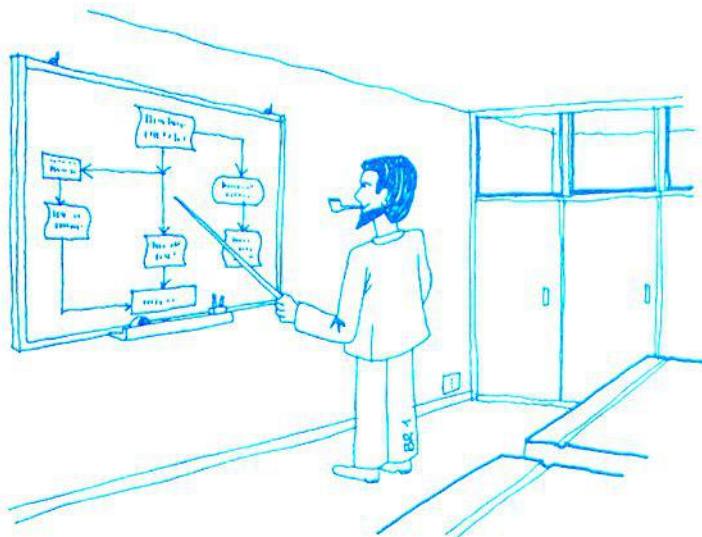
Nè un esperto programmatore in Turbo C deve sentirsi superiore solo perchè riesce a scrivere programmi più veloci di un Basicomane; quest'ultimo, infatti, potrebbe far ricorso ad un semplice compilatore Basic (o, in ambiente Ms-Dos, al velocissimo QuickBasic) per battere in velocità, più spesso di quanto non si creda, un programma scritto in C.

Continua quindi a valere la regola "Meglio un programm efficiente in un linguaggio interpretato che uno inefficiente in uno compilato".

E allora? Che senso può avere dedicare tempo e pazienza (tantissima) per imparare le regole di un nuovo linguaggio che, alla fine, fa tutte le cose che fa il nostro Basic che abbiamo imparato con tanta fatica?

La risposta è identica a quella che verrebbe data a chi, ingenuamente, chiedesse "A che serve imparare l'inglese se già conosco l'italiano?".

La necessità di imparare un altro linguaggio, pertanto, è strettamente legata al mondo culturale nel quale viviamo. Se negli anni '50 era sufficiente conoscere solo l'italiano (anche sommariamente), oggi è praticamente impossibile aspirare ad un certo "ambiente" (in senso lato) ignorando l'inglese; in certi ambienti, addirittura, non è sufficiente la conoscenza di due lingue straniere.

E così anche nel campo dell'informatica riteniamo che i tempi siano ormai maturi per dedicarsi allo studio di un secondo (terzo?) linguaggio, per diletto o per aggiornamento personale; per non restare ancorati eternamente, insomma, all'età della pietra-Basic.

duttiva appare subito antieconomica (si vende senza guadagnare nulla o, peggio, sottocosto). In casi come questi, si potrà obiettare, non vale neanche la pena far ricorso al computer. Per completezza, tuttavia, abbiamo pensato anche agli Input senza... senso.

Il messaggio "Antieconomico" (riga 250), in definitiva, verrà visualizzato se il ricavo è minore del costo (riga 160). Nel caso in cui i tre parametri di Input (CF, CU, RU) siano nulli (caso banale) il messaggio più opportuno sarà "Non problema" che è un modo eufemistico di dire "Perchè usi un computer in modo così scemo?". La stessa frase apparirà nel caso in cui (riga 190) il ricavo è maggiore del costo, ma le spese fisse sono nulle: non è certo necessario il computer per sapere che c'è un certo vantaggio economico.

In un solo caso, quindi, appare una risposta "decente"; ci riferiamo, cioè, al caso in cui CF esiste e RU risulta maggiore di CU. Verrà visualizzato, in questo caso, il numero di oggetti da produrre per raggiungere in Breakevenpoint.

Vogliamo attirare l'attenzione del lettore sul fatto che i "filtri" software sono stati posti in un ordine di elaborazione tale per cui questa viene interrotta al verificarsi di condizioni troppo banali. Ci riferiamo, in particolare, al caso dei tre Input nulli (riga 150); subito dopo, come ordine di banalità, viene attivato il filtro di riga 160 che dovrebbe ricordare all'utente l'inutilità di elaborare una situazione immediatamente riconoscibile come antieconomica.

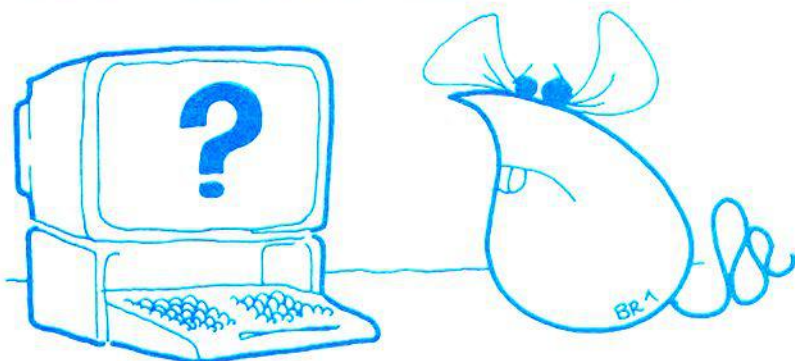
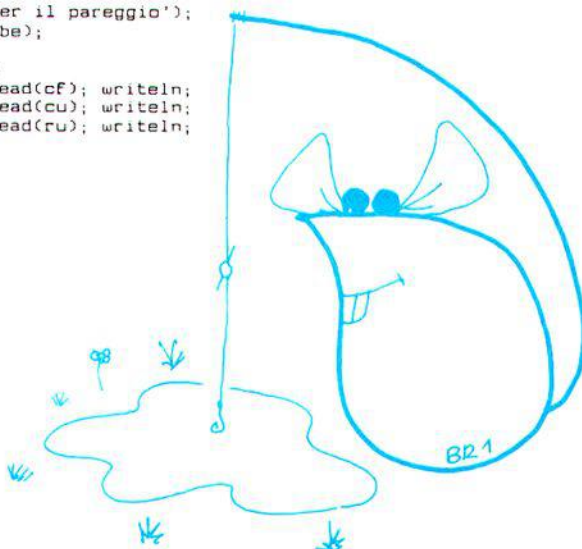
Allo stesso modo la riga 190 impedisce di proseguire il calcolo perchè anche in questo caso non è certo necessario un computer per risolvere il problema.

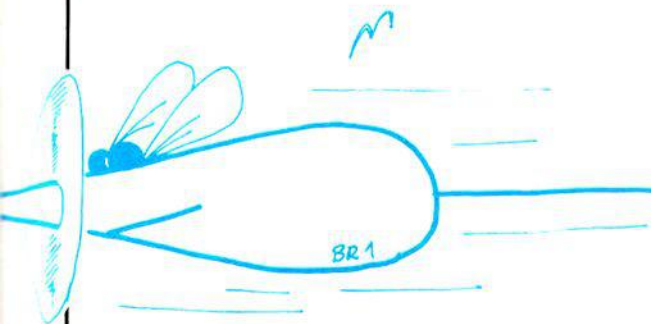
Si sottolinea il fatto che una terza ipotesi (relativa ad Input negativi) è stata volutamente trascurata non soltanto perchè ritenuta banale o per sollecitare il lettore a studiarne un eventuale "trattamento". Potrebbe infatti esistere convenienza a produrre in perdita, almeno per un periodo limitato, disponendo di apparecchiature gratuite (CF minore di zero) da rivendere dopo un certo periodo produttivo. Analogamente potrebbe verificarsi il caso, pur se particolare, di un valore CU negativo in cui il materiale da lavorare sia fornito a pagamento: si pensi ai rifiuti tossici, che vengono ceduti non solo gratis e a domicilio, ma addirittura accompagnati da forti somme di denaro.

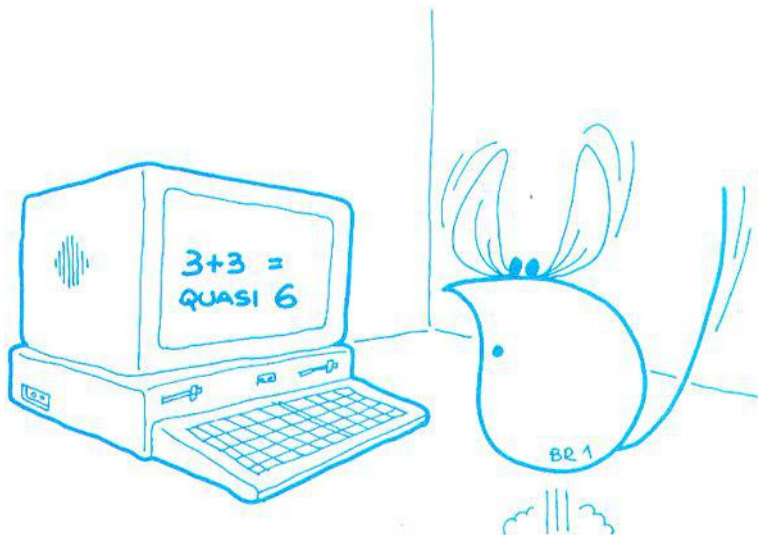
IL PROGRAMMA IN TURBO PASCAL

Chi non è avvezzo al Pascal sicuramente storcerà il naso confrontando tra loro le lunghezze dei due listati Basic e Pascal. Probabilmente, anche ad un esame superficiale, il programma in Pascal può apparire ridondante e pieno di istruzioni superflue se non inutili.

Confrontando più attentamente i due listati, tuttavia, non si può negare una maggiore "scorrevolezza" del Pascal rispetto al Basic. Per individuare il funzionamento del programma, infatti, è necessario uno "studio" della versione Basic, mentre, per quella in Pascal, può essere sufficiente una sua "lettura".

Si noti, tra l'altro, la necessità (pena segnalazione di errore) di indicare all'inizio i nomi e la tipologia delle variabili che verranno adoperate; in Basic, al contrario, qualunque varia-

```
program BREAKEVENPOINT;
var cf,cu,ru,be:real;
procedure antieconomico; (* prima   "subroutine" *)
      begin
      writeln('antieconomico!');
      end;
procedure nonproblema;    (* seconda "subroutine" *)
      begin
      writeln('non - problema!');
      end;
procedure messaggio;      (* terza   "subroutine" *)
      begin
      write ('n. unita'' da produrre per il pareggio');
      write (' di ricavi - spese = ', be);
      end;
BEGIN (*INIZIO PROGRAMMA PRINCIPALE*)
      write ('costo fisso? ');          read(cf); writeln;
      write ('costo per unita''? '); read(cu); writeln;
      write ('ricavo per unita''? ');read(ru); writeln;
if    (cf=0) and (ru=0) and (cu=0)
then BEGIN
      nonproblema
      END
else BEGIN
      if    (ru<=cu)
      then BEGIN
            antieconomico
            END
      else BEGIN
            be:=(cf/(ru-cu));
            if (be=0)
            then BEGIN
                  nonproblema
                  END
            else BEGIN
                  messaggio
                  END
            END
      END
END.  (*FINE PROGRAMMA *)
```

```
program BREAKEVENPOINT; var cf,cu,ru,be:real;
procedure antieconomico; begin writeln('antieconomico!'); end; procedure
nonproblema; begin writeln('non - problema!'); end; procedure messaggio;
begin write ('n. unita'' da produrre per il pareggio'); write ('ric-sp',
be); end; BEGIN write ('costo fisso? '); read(cf); writeln;
write ('costo per unita''? '); read(cu); writeln;write ('ricavo per unita''? ');
read(ru); writeln; if (cf=0) and (ru=0) and (cu=0) then BEGIN
nonproblema END else BEGIN if (ru<=cu) then BEGIN antieconomico END else BEGIN
be:=(cf/(ru-cu)); if (be=0) then BEGIN nonproblema END else BEGIN messaggio END
END END END.
```

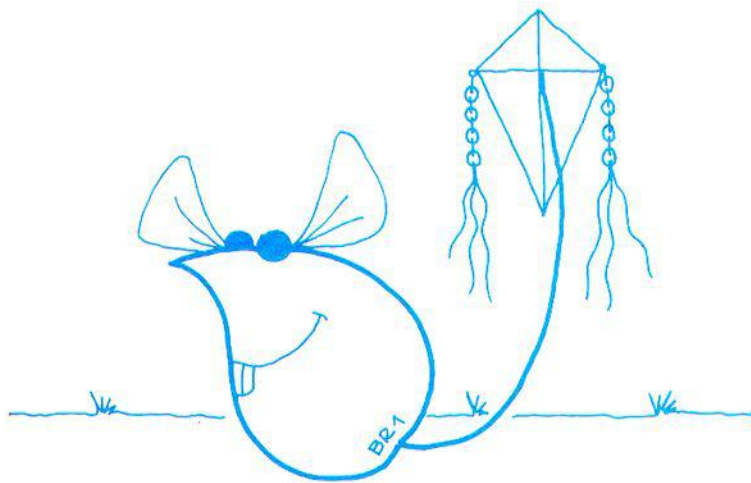
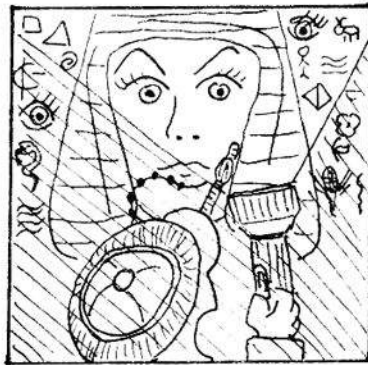
```
100 REM PROGRAMMA PER DETERMINARE
110 REM IL BREAKEVENPOINT
111 :
120 INPUT "COSTO FISSO";CF
130 INPUT "COSTO PER UNITA'";CU
140 INPUT "RICAVO PER UNITA'";RU
150 IF CF=0 AND CU=0 AND RU=0 THEN 260
160 IF RU <= CU THEN 250
180 BE = CF / (RU-CU)
190 IF BE=0 THEN 260
210 PRINT "N. UNITA' DA PRODURRE"
220 PRINT "PER IL PAREGGIO DI"
230 PRINT "RICAVI - SPESE ="BE
240 END
250 PRINT "ANTIECONOMICO":END
260 PRINT "NON - PROBLEMA":END
```

bile può essere definita in qualsiasi momento e luogo creando, di conseguenza, difficoltà di ricerca. Allo stesso modo, in Pascal, la necessità di raggruppare tutte le Procedure (una sorta di subroutine del Basic) in una parte standard del programma, si rivela comodissima per individuarle con facilità in qualsiasi momento. Per non parlare della "indentazione", termine che indica la possibilità (attivata subito, come default, al caricamento del Turbo Pascal) di allineare verticalmente le varie istruzioni in modo da realizzare una struttura schematica di notevole efficacia in termini di comprensione del listato stesso.

Per ciò che riguarda il listato specifico di queste pagine, la Procedura "Antieconomico", come le altre due, possono apparire inefficienti ma consentono, al lettore desideroso di apprendere i rudimenti del Pascal, di capire in che modo una Procedura può essere richiamata durante l'elaborazione del programma. Analogamente il lettore, sfruttando l'indentazione utilizzata, può notare l'efficacia visiva dell'allineamento verticale delle strutture If.. Then..Else come pure la corrispondenza, sempre verticale, delle strutture Begin...End. Ad ogni Begin corrisponde il "suo" End e non c'è possibilità di sbagliare; in ogni caso ci pensa il Pascal a segnalare la mancanza di qualche End.

Altrettanto importante è la punteggiatura; si noti l'ubicazione dei vari caratteri di punto e virgola (;); la corrispondenza tra il Print del Basic e il Write e Writeln del Pascal; come pure tra Input (Basic) e Read (Pascal); la necessità di inserire parentesi; la possibilità di inserire commenti; il massiccio ricorso a Begin...End per evitare, soprattutto nelle strutture If..Then...Else, di effettuare elaborazioni indesiderate.

Teniamo a sottolineare ancora una volta l'importanza della struttura Begin...End, sulla quale si fonda il linguaggio stesso: meglio un Begin...End in più che uno in meno; male che vada, infatti, viene ignorato.

Non si pensi, tuttavia, che in Pascal sia impossibile scrivere "male" un programma; in queste pagine, infatti, è riportata anche una malaversione scritta "di seguito" che, a meno dei commenti, è rigorosamente identica a quella indentata, leggibilità a parte.

A questo punto chiunque abbia seguito con attenzione le note precedenti dovrebbe essere in grado di sviluppare una (quasi) notevole varietà di programmi in Pascal dal momento che, nel programma di esempio, sono state impiegate le istruzioni di uso più frequente in qualsiasi linguaggio: definizione variabili, ingresso dati, visualizzazioni, elaborazioni condizionate (con alternative: Else), subroutine.

Prossimamente esamineremo, un po' per volta, le notevoli potenzialità offerte da quello straordinario compilatore che risponde al nome di Turbo Pascal.

EASY LIST

Non è più necessario ricorrere alla stampante per esaminare con facilità un listato Basic

di **Fabio Sorgato**

Esaminare il listato di un programma Basic che occupi più di una decina di righe, direttamente sul video, richiede una pazienza da certosino.

Quasi sempre, infatti, i programmi sono ben imbottiti di istruzioni Goto e Gosub che ci costringono ad una visualizzazione enorme di List con pressioni abili e repentine di Run / Stop e di Ctrl per evitare che la prima riga della subrouitine finisca fuori dal video.

Chi è in possesso di una stampante preferisce, di conseguenza, stampare il listato su carta in modo da spaziare liberamente da un punto all'altro del programma; chi, purtroppo, è sprovvisto della scrivente si arrende oppure rischia di lavorare con i nervi a fior di pelle.

La routine qui pubblicata, lunga poche(!) centinaia di byte, permette di gestire un listato Basic come se fosse stampato su un foglio di carta, con possibilità di muoversi su e giù lungo il listato con estrema facilità.

COME SI USA

Anche chi di LM è completamente digiuno può servirsi di questa routine: basta copiare il programma caricatore (quello cioè, contenente le righe Data), salvarlo per sicurezza su supporto magnetico, e dare Run.

A questo punto verrà richiesto dove allocare la routine LM; l'area consigliata è quella posizionata a partire da 49152 perchè si evita, in tal modo, di ridurre lo spazio di memoria riservata ai programmi Basic.

Il programma comunicherà come riabilitare la routine dopo averla eventualmente disabilitata con Run Stop / Restore.

Per controllare che tutto sia in ordine scrivete il numero uno (1) e (senza premere il tasto Return) tenete premuto il tasto CRSR down: dovreste vedere scorrere il listato del programma in memoria; premete quindi lo stesso tasto ma con lo shift (CRSR up). In questo caso il programma scorrerà dall'alto verso il basso, mostrando di riga in riga le linee precedenti a quelle sul video.

Insomma è come se aveste a che fare con un word processor che vi consente di andare ovunque nel testo mediante la semplice pressione di crsr down e crsr up!

Potete ora caricare il programma da esaminare e farlo "scorrere" avanti e indietro con enorme facilità semplicemante digitando un numero di linea e premendo CRSR up oppure down.

Il programma, infatti, smette di funzionare in due casi: se si premono i tasti Run/Stop e Restore (dal momento che funziona in interrupt) e se si cancella lo schermo, o meglio se sul video non compare alcun valore numerico compatibile con la numerazione Basic.

Per evitare di scorrere l'intero programma, per raggiungere una particolare linea è sufficiente scriverne il numero di linea (diminuito di uno) sullo schermo "vuotato" con Shift + Clr/Home; poi premere CRSR down.

COME FUNZIONA

Il disassemblato commentato è riportato, come al solito, per i più esperti; comunque vale la pena riassumere a blocchi i vari compiti svolti dalla routine.

Tramite le istruzioni Poke, che abilitano Easy List, vengono modificati i puntatori della routine di decodifica del tasto premuto ($028F - $0290) in modo da saltare all'inizio della nostra routine; qui dapprima si decodifica il tasto tramite la normale routine ($EB48) e vengono eseguiti i primi controlli: se sta girando un programma (run mode) e se il tasto premuto non è CRSR up nè CRSR down, torna alle funzioni normali.

Viene quindi spento il cursore per evitare che una cifra del nu-

mero di linea visualizzato sullo schermo rimanga in reverse.

Se è stato premuto CRSR down, ma il cursore non è sull'ultima linea, torna alle funzioni normali, altrimenti cerca la prima linea, partendo dal basso, che inizi con un carattere numerico; se nessuna linea inizia con un carattere numerico, esce, altrimenti salta la routine di CRSR up.

Se è stato premuto CRSR up, ed il cursore non è sulla prima linea di schermo, esce, altrimenti cerca la prima linea che inizi con un carattere numerico partendo dall'alto e, se non la trova, esce.

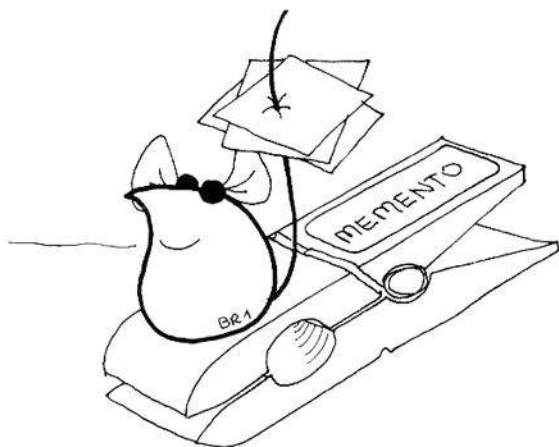
C'è un'altra parte comune a CRSR up e down: quella che elimina il repeat per evitare interferenze con la routine, converte il valore trovato in un intero e fa puntare i vettori dell'error message ($0300 - $0301) ad un RTS per non fare stampare "READY." dopo il list di ogni linea.

Se la routine è quella per il CRSR down, incrementa il numero di linea, cerca la linea a partire da quel numero tramite la $A613, controlla che non sia terminato il programma, stampa un RETURN, esegue il list della linea, aggiusta i puntatori ed esce.

Se è stato invece premuto CRSR up, cerca la linea precedente al numero trovato sul video, controllando che esista il programma e che la linea non sia la prima del programma, inserisce una linea di 40 caratteri in alto allo schermo, imposta l'auto scroll mode in modo che se la linea da stampare è più lunga di 40 caratteri, il SO provveda automaticamente ad inserire un'altra linea; stampa un HOME, esegue il list della linea, aggiusta i puntatori ed esce.

ALCUNI TRUCCHI USATI

Per creare una routine rilocabile e breve abbiamo più volte dovuto ingannare il sistema operativo o ricorrere ad alcuni trucchetti che ora spiegheremo per rendere più comprensibile il funzionamento.

1 - Spegnere il cursore: è stato necessario spegnere il cursore perchè se si fosse trovato acceso sopra una cifra del numero di linea avrebbe impedito la conversione del numero tramite la routine di VAL; per spegnerlo abbiamo utilizzato il flag del cursore ($CF) che contiene 1 quando il cursore è acceso e 0 quando è spento, tramite uno shift binario a destra il bit 0 viene fatto cadere nel carry, e con una successiva rotazione dal carry viene spostato nel bit 7 che, guarda caso è quello che indica il reverse di un carattere. A questo punto siamo ricorsi ai puntatori della linea del cursore ($D1 - $D2) e al puntatore alla colonna ($D3), eseguendo un or esclusivo (EOR) tra il contenuto della locazione di memoria di schermo dove è localizzato il cursore e il valore ottenuto dalle rotazioni precedenti, si ha il seguente risultato:

Primo caso: se il cursore è spento (byte ottenuto dalla rotazione = %00000000) e risulta che...

...il carattere è normale = nessun effetto
...il carattere è reverse = nessun effetto

Se il cursore è acceso (byte ottenuto dalla rotazione = %10000000) e risulta che...

...il carattere è normale = il carattere diventa reverse
...il carattere è reverse = il carattere diventa normale.

2-Convertire il numero da codice video in numero intero senza segno in $14 - $15.

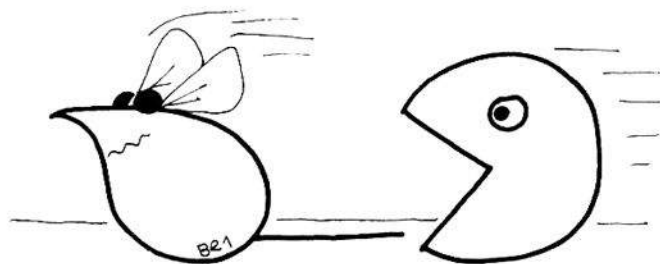
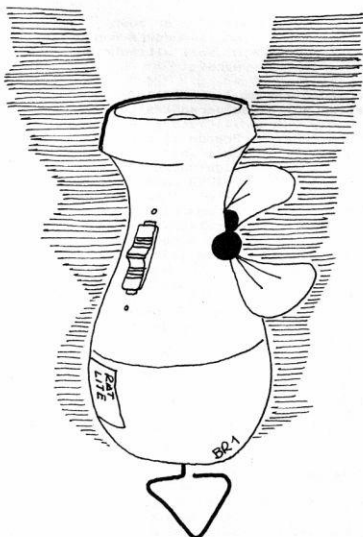
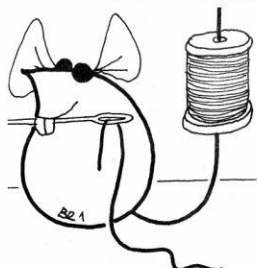
Abbiamo in questo caso utilizzato la routine del SO che esegue la VAL del Basic facendogli "credere" che il numero sullo schermo sia una stringa di lunghezza 5 di cui eseguire VAL, con successiva conversione in intero tramite la $B7F7 (che segnala illegal quantity nel caso in cui il numero di linea sia superiore a 65535).

Abbiamo potuto utilizzare questo stratagemma grazie all'eguaglianza dei codici dei numeri in Ascii e in memoria video.

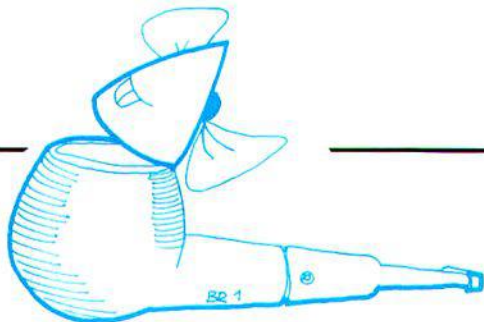
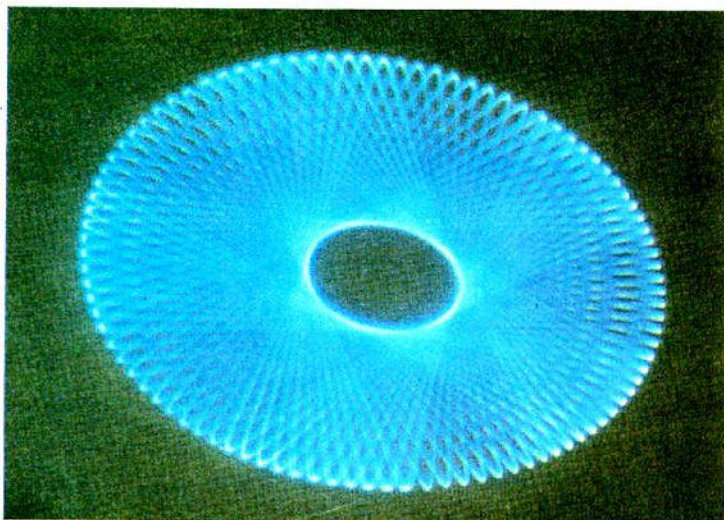
3-Utilizzare la list del so.

Abbiamo dovuto ricorrere a vari accorgimenti per evitare di riscrivere una routine di list (con conseguente allungamento della routine).

Dal momento che List termina con un JMP alla routine che stampa gli errori (con l'ordine di stampare "READY."), abbiamo dovuto modificare i puntatori della error routine ($0300 - $0301) facendola puntare a $B9BB che contiene un RTS, in modo che dopo aver Listato la linea, il controllo ritorni alla easy list.

Siccome, poi, la routine List stampa un return dopo avere visualizzato la linea Basic (a meno che questa non sia l'ultima linea del programma), abbiamo dovuto simulare una fine programma azzerando il byte alto del link della linea successiva.

```
10 REM ** EASY LIST RILOCABILE
20 REM ** PER C/64
30 REM ** BY FABIO SORGATO
40 REM
50 PRINT CHR$(147);TAB(23);"49152";CHR$(19);
60 INPUT"DOVE METTO LA ROUTINE";S
70 IF S<=PEEK(45)+PEEK(46)*256 THEN 80
75 IF (S>2047 AND S<40615) OR (S>49151 AND S<52903) THEN 90
80 PRINT"AREA GIA' OCCUPATA DAL SISTEMA":END
90 IF PEEK(55)+PEEK(56)*256<S THEN 100
95 POKE 56,(S-1)/256:POKE 55,(S-1)-256*PEEK(56)
100 CK=0:FOR K=0 TO 345
110 READ A:CK=CK+A
120 POKE S+K,A:NEXT
121 IF CK<>47180 THEN PRINT"ERRORE NEI DATI !!":END
125 PRINT"PER DISABILITARE: RUN STOP/RESTORE":PRINT
130 PRINT"PER ABILITARE EASY LIST:":PRINT
140 PRINT"POKE 655,";S-INT(S/256)*256;
150 PRINT":POKE 656,";INT(S/256)
160 POKE 198,4:POKE631,145:POKE632,145:POKE633,145:POKE634,13:REM ABILITA
170 :
1000 DATA 032,072,235,164,157,240,040
1010 DATA 189,118,002,133,250,201,017
1020 DATA 240,004,201,145,208,027,164
1030 DATA 211,165,207,074,106,081,209
1040 DATA 145,209,160,000,132,207,200
1050 DATA 132,204,160,000,165,250,201
1060 DATA 017,240,003,208,038,096,166
1070 DATA 214,224,024,208,249,181,217
1080 DATA 044,067,167,240,019,041,127
1090 DATA 133,252,189,240,236,133,251
1100 DATA 177,251,201,048,144,004,201
1110 DATA 058,144,042,202,016,227,048
1120 DATA 218,166,214,208,214,181,217
1130 DATA 044,067,167,240,019,041,127
1140 DATA 133,252,189,240,236,133,251
1150 DATA 177,251,201,048,144,004,201
1160 DATA 058,144,007,232,224,025,208
1170 DATA 225,240,181,173,138,002,072
1180 DATA 169,064,141,138,002,165,251
1190 DATA 133,034,165,252,133,035,160
1200 DATA 005,152,032,181,183,032,247
1210 DATA 183,173,000,003,072,173,001
1220 DATA 003,072,169,187,141,000,003
1230 DATA 169,185,141,001,003,165,250
1240 DATA 201,017,208,084,230,020,208
1250 DATA 002,230,021,032,019,166,160
1260 DATA 001,177,095,133,254,240,052
1270 DATA 169,019,166,250,224,017,208
1280 DATA 002,169,013,032,210,255,169
1290 DATA 000,133,198,136,177,095,133
1300 DATA 253,200,177,253,072,169,000
1310 DATA 145,253,200,177,095,133,020
1320 DATA 200,177,095,133,021,160,001
1330 DATA 132,015,032,215,166,160,001
1340 DATA 104,145,253,104,141,001,003
1350 DATA 104,141,000,003,104,141,138
1360 DATA 002,096,208,181,032,019,166
1370 DATA 165,096,197,044,208,006,165
1380 DATA 095,197,043,240,226,165,043
1390 DATA 133,251,165,044,133,252,160
1400 DATA 001,177,251,240,212,197,096
1410 DATA 208,007,136,177,251,197,095
1420 DATA 240,014,160,000,177,251,170
1430 DATA 200,177,251,133,252,134,251
1440 DATA 208,225,165,251,133,095,165
1450 DATA 252,133,096,162,000,032,104
1460 DATA 233,173,136,002,009,128,133
1470 DATA 217,162,000,134,214,134,211
1480 DATA 032,108,229,169,001,141,146
1490 DATA 002,208,164
1500 END
```

"EASY LIST" - DISASSEMBLATO COMMENTATO

```
       JSR $EB48    ;Decodifica il tasto
                    ;premuto
       LDY $9D      ;Se il flag indica
       BEQ XX02F    ;run mode, esce.
       LDA $0276,X  ;Salva il codice
       STA $FA      ;del tasto.
       CMP #$11     ;Se non e' crsr down
       BEQ XX014    ;
       CMP #$91     ;e se non e' crsr up,
       BNE XX02F    ;esce.
XX014  LDY $D3      ;Se il cursore
       LDA $CF      ;e' acceso, lo spegne
       LSR A        ;
       ROR A        ;
       EOR ($D1),Y  ;
       STA ($D1),Y  ;
       LDY #$00     ;Dispone il flag
       STY $CF      ;cursore spento.
       INY          ;Inibisce il lampeg-
       STY $CC      ;gio del cursore.
       LDY #$00     ;
       LDA $FA      ;
       CMP #$11     ;Se non e' stato
       BEQ XX030    ;premuto crsr down,
       BNE XX055    ;esegue la routine
                    ;di crsr up.
XX02F  RTS          ;Esce.
;
;Routine per crsr down
;
XX030  LDX $D6      ;Se e' crsr down
       CPX #$18     ;ma il cursore non
       BNE XX02F    ;e' sull'ultima riga,
                    ;esce.
;
;Cerca un numero di linea sullo schermo
;a partire dal basso verso l'alto
;
XX036  LDA $D9,X    ;Prende il byte alto
                    ;dell'inizio della
                    ;riga corrente.
       BIT $A743    ;($A743 contiene $80)
                    ;se e' il continuo
       BEQ XX050    ;di una riga a 80
                    ;caratteri, passa
                    ;alla successiva.
       AND #$7F     ;Salva il byte alto
       STA $FC      ;in $FC
       LDA $ECF0,X  ;e il byte basso
       STA $FB      ;in $FB.
       LDA ($FB),Y  ;Legge il primo ca-
                    ;rattere della riga,
       CMP #$30     ;
       BCC XX050    ;
       CMP #$3A     ;
       BCC XX07A    ;se non e' numerico
XX050  DEX          ;passa alla riga
       BPL XX036    ;precedente;
       BMI XX02F    ;se finscono le
                    ;righe, esce.
;
;Routine per crsr up
;
XX055  LDX $D6      ;Se e' stato premuto
                    ;crsr up e' il cur-
       BNE XX02F    ;sore non e' sulla
                    ;prima riga, esce.
```

```
;Cerca una numero di linea sullo
;schermo, dall'alto verso il basso
;
XX059  LDA $D9,X    ;Legge il byte alto
                    ;della riga corrente
       BIT $A743    ;Se e' il continuo
                    ;di una riga a 80
       BEQ XX073    ;caratteri, passa
                    ;alla prossima linea.
       AND #$7F     ;Salva il byte alto.
       STA $FC      ;in $FC,
       LDA $ECF0,X  ;mette il byte basso
       STA $FB      ;in $FB.
       LDA ($FB),Y  ;Se il primo carat-
       CMP #$30     ;tere della riga
       BCC XX073    ;non e' numerico,
       CMP #$3A     ;
       BCC XX07A    ;
XX073  INX          ;passa alla riga
                    ;successiva.
       CPX #$19     ;Se restano ancora
       BNE XX059    ;righe, ripete,
       BEQ XX02F    ;Altrimenti esce.
;
;converte il valore sul video in
;un intero.
;
XX07A  LDA $028A    ;Salva il valore
       PHA          ;del repeat flag
       LDA #$40     ;e toglie il repeat
       STA $028A    ;a tutti i tasti.
       LDA $FB      ;Crea una stringa
       STA $22      ;fasulla a partire
       LDA $FC      ;dal primo carattere
       STA $23      ;di una riga di
                    ;schermo,
       LDY #$05     ;con lunghezza 5.
       TYA          ;
       JSR $B785    ;Esegue VAL,
       JSR $B7F7    ;converte in intero
                    ; in $14-$15          ;
;aggiusta i puntatori
;
       LDA $0300    ;Salva i puntatori
       PHA          ;della routine di
       LDA $0301    ;errore (che stampa
       PHA          ;anche READY.) nello
                    ;stack
       LDA #$E8     ;e punta ad un RTS.
       STA $0300    ;
```

```
       LDA #$89     ;
       STA $0301    ;
       LDA $FA      ;Se il tasto era
       CMP #$11     ;crsr down, esegue
       BNE XX100    ;la routine.
;
;routine per il crsr down
;
       INC $14      ;Aggiunge 1 al valore
       BNE XX0B2    ;trovato sullo
       INC $15      ;schermo.
XX0B2  JSR $A613    ;Cerca la linea basic
;
;list della linea
;
XX0B5  LDY #$01     ;Sposta il byte alto
       LDA ($5F),Y  ;del link in $FE,
       STA $FE      ;
       BEQ XX0F1    ;se e' 0, sono termi-
                    ;nate le linee di
                    ;programma: esce.
       LDA #$13     ;Se non e' stato
                    ;chiamato dalla rou
       LDX $FA      ;ine di crsr down,
       CPX #$11     ;stampa HOME
       BNE XX0C7    ;altrimenti stampa
       LDA #$0D     ;RETURN.
XX0C7  JSR $FFD2    ;
       LDA #$00     ;Azzera il buffer
       STA $C6      ;di tastiera.
       DEY          ;Salva il byte basso
       LDA ($5F),Y  ;del link in $FD.
       STA $FD      ;
       INY          ;
       LDA ($FD),Y  ;Conserva il byte al-
       PHA          ;to del link della
                    ;linea successiva
                    ;nello stack
       LDA #$00     ;e crea una fine
       STA ($FD),Y  ;programma fasulla.
       INY          ;
       LDA ($5F),Y  ;Mette il numero
       STA $14      ;della linea
       INY          ;da listare,
       LDA ($5F),Y  ;in $14-$15.
       STA $15      ;
       LDY #$01     ;Flag = non stringa.
       STY $0F      ;
       JSR $A6D7    ;Esegue list.
;
;Ripristina puntatori ed esce
;
       LDY #$01     ;Ripristina il valore
       PLA          ;sostituito in prece-
       STA ($FD),Y  ;denza con uno zero.
XX0F1  PLA          ;Ripristina i valori
       STA $0301    ;dei puntatori
       PLA          ;alla routine di
       STA $0300    ;errore,
       PLA          ;e il valore
       STA $028A    ;del repeat flag.
       RTS          ;Esce.
;
XX0FE  BNE XX0B5    ;Richiamo del BNE
                    ;a fine programma
;
;routine per il crsr up
;
XX100  JSR $A613    ;Cerca la linea
       LDA $60      ;nel programma,
       CMP $2C      ;se e' la prima linea
       BNE XX10F    ;o se non c'e' nessun
       LDA $5F      ;programma,
       CMP $2B      ;
       BEQ XX0F1    ;esce.
;
;Cerca la linea precedente, cioe'
;quella il cui link punta alla
;linea attuale
;
XX10F  LDA $2B      ;Trasferisce i punta-
       STA $FB      ;tori di inizio pro-
       LDA $2C      ;gramma basic
       STA $FC      ;in $FB-$FC.
XX117  LDY #$01     ;Se e' la fine
       LDA ($FB),Y  ;programma,
       BEQ XX0F1    ;esce.
       CMP $60      ;Confronta il byte
                    ;alto della linea in
                    ;esame con quello
                    ;della linea corrente
       BNE XX128    ;
       DEY          ;se sono uguali con-
       LDA ($FB),Y  ;fronta i byte bassi.
       CMP $5F      ;Se sono uguali,
       BEQ XX136    ;linea trovata.
XX128  LDY #$00     ;Passa alla
       LDA ($FB),Y  ;linea successiva.
       TAX          ;
       INY          ;
       LDA ($FB),Y  ;
       STA $FC      ;
       STX $FB      ;
       BNE XX117    ;
XX136  LDA $FB      ;Salva la locazione
       STA $5F      ;di inizio della
       LDA $FC      ;riga basic
       STA $60      ;in $5F-$60
;
;inserisce una linea di 40 caratteri
;in alto allo schemo e imposta
;l'auto scroll mode
;
       LDX #$00     ;Inserisce una riga
       JSR $E968    ;in alto allo
                    ;schermo.
       LDA $0288    ;La linea inserita e'
       ORA #$80     ;di 40 caratteri.
       STA $D9      ;
       LDX #$00     ;Aggiorna gli
       STX $D6      ;indirizzi della
       STX $D3      ;linea 0 dello
       JSR $E56C    ;schermo.
       LDA #$01     ;Imposta l'auto
       STA $0292    ;scroll.
       BNE XX0FE    ;JMP alla list.
```

COLORI E NUMERI

Due nuove routine di utilità generale per arricchire la vostra enciclopedia di programmi universali

Gli Amici Di Amiga

Questo mese presentiamo due semplici sub-routine che permettono di colorare lo schermo servendosi dei nomi stessi dei colori e di far apparire cifre numeriche grandi a volontà.

SETCOLOR

Questa funzione permette di assegnare, ai registri colore di Amiga, i valori opportuni usando la stringa col nome esplicito (ROSSO, BLUSCURO), invece che un complesso comando PALETTE con oscuri parametri numerici. Ciascuno può dunque fabbricarsi la sua libreria di colori e tonalità, facilmente riusabili in tutti i programmi.

I formati per "chiamare" Setcolor sono...

CALL Setcolor (a%, c$)
Setcolor a%, c$

...in cui A% è un parametro intero (numero o variabile) che specifica il numero di registro di colore (come nel comando PALETTE), mentre C$ è una stringa di caratteri indifferentemente maiuscoli e/o maiuscoli che indicano il colore.

La funzione ricerca nella tabella dei DATA, posta in coda alla funzione, la stringa equivalente a quella indicata e, se trovata, legge le tre componenti di colore Rosso, Blu e Giallo e le assegna al numero di registro (tipicamente 0 ... 3) specificato da a%.

Per inserire altri colori in coda alla funzione (tra l'ultimo DATA del listato pubblicato ed il comando END SUB) il formato deve essere...

DATA NOMECOLORE, r, g, b

...in cui "NOMECOLORE" è la stringa col nome del colore, in caratteri tutti maiuscoli, mentre i tre parametri numerici successivi sono esattamente come per un comando PALETTE (vedi manuale Amigabasic pagina 8 - 104)

Aggiungendo altri colori, bisogna incrementare concordemente il numero 14 del ciclo FOR ... NEXT, affinchè consulti anche le nuove frasi DATA.

La funzione prevede anche di riassegnare in ogni istante la paletta dei primi quattro registri di colore ai valori standard iniziali, semplicemente indicando come colore la stringa "STANDARD", indipendentemente dal numero espresso come registro. In questo caso, infatti, vengono alterati i primi quattro contemporaneamente.

DISPLAYLED

Questa funzione permette di stampare cifre colorate di dimensioni volute in un formato "a sette segmenti", tipo display luminoso a LED. I modi per chiamare la funzione da un nostro programma sono i due seguenti:

CALL DisplayLED(n$, x%, y%, g%)
DisplayLED n$, x%, y%, g%

I parametri hanno i sueguenti significati:

n$:deve essere una variabile stringa che contiene i numeri da visualizzare. La lunghezza massima è data dalla dimensione scelta per i caratteri e dalla posizione sul video.
x% :deve essere un numero od una variabile intera che specifichi la posizione oriz-

zontale (rispetto all'origine dello schermo attuale) dove inizierà la scrittura della stringa.
y% :come il precedente, ma per la posizione verticale. Ovviamente x% e y% possono essere compresi tra zero e le massime dimensioni concesse dalle risoluzioni grafiche o dalle dimensioni di finestra impostate.
g% :Indica l'altezza delle cifre, misurata in pixel. Si consiglia di non scendere al di sotto dei quattro pixel.

Il numero gigante viene visualizzato sfruttando il colore numero tre della paletta attuale, tipicamente arancione.

COME REGISTRARE

Ricordiamo che la procedura per salvare, su disco, il sotto-programma in una nostra libreria è la seguente:

1) Tagliare (opzione CUT) le linee di programma esemplificative che servono solo a "collaudare" la funzione vera e propria pubblicata in queste pagine.
2) Inserire il dischetto con la nostra directory di libreria in un drive e si registri la funzione come file ASCII. Supponendo che il disco in questione sia nel drive interno e che la directory delle funzioni di libreria si chiami "libraryccc" useremo:

```
SAVE "df0:libraryccc/DisplayLED",a
```

In questo modo, quando vorremo inserire la funzione in un altro programma, supponendo di avere inserito il disco nel primo drive esterno, useremo semplicemente, dal modo di comandi diretti:

```
MERGE "df1:libraryccc/DisplayLED"
```

Si ricordi che un file è inseribile con MERGE solo quando è stato registrato in formato ASCII.

L'ENCICLOPEDIA DI ROUTINE IN AMIGABASIC

Tra le tante possibilità offerte dall'AmigaBasic esiste anche quella di creare sottoprogrammi che sono un insieme di istruzioni, atte a svolgere una particolare funzione, richiamabili da qualsiasi parte del programma. Tale meccanismo di richiamo evita di scrivere la stessa sequenza di istruzioni ogni volta che serve quella particolare funzione.

Le subroutine, che si attivano con un GOSUB, terminano con un RETURN, in modo che il controllo passi alla istruzione successiva al Gosub chiamante.

Il meccanismo di richiamo è valido anche per i sottoprogrammi (a parte le parole - chiave) ma la grande differenza risiede nella gestione delle variabili. Come dice il nome stesso, infatti, un sottoprogramma è un programa a sè stante con le proprie variabili, area di memoria e così via, ma si trova esso stesso all'interno di un programma. Ciò significa che possiamo organizzare le variabili all'interno di un sottoprogramma in maniera molto libera: tenendo conto delle variabili esterne, non tenendone conto affatto, oppure tenendone conto solo parzialmente. La procedura si concretizza nella più assoluta indipendenza del sottoprograma che, quindi, ben si presta per realizzare routine di utilità generale richiamabili da vari programmi. Vediamo ora come si dichiara un sottoprogramma:

```
SUB Nome Sub Program (parametri) STATIC
Istruzioni del sottoprogramma
END SUB
```

Esso può assumere una posizione qualsiasi (ma preferibilmente al fondo) nel programma in quanto non potrà mai essere coinvolto nel flusso di esecuzione: viene semplicemente ignorato, a meno che non facciamo una esplicita richiesta con CALL che, in questo frangente, ha la seguente sintassi...

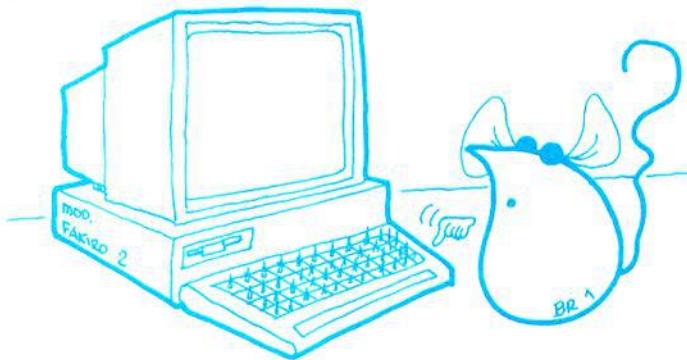
```
CALL Nome Sub Program (parametri)
```

L'unico "ponte" di comunicazione tra programma e sottoprogramma è rappresentato dai parametri, vale a dire le variabili (o costanti) che intendiamo "passare". Se dichiariamo un certo numero di variabili come parametri in entrata, dovremo effettivamente fornire quello stesso numero di variabili quando effettuiamo la chiamata (CALL).

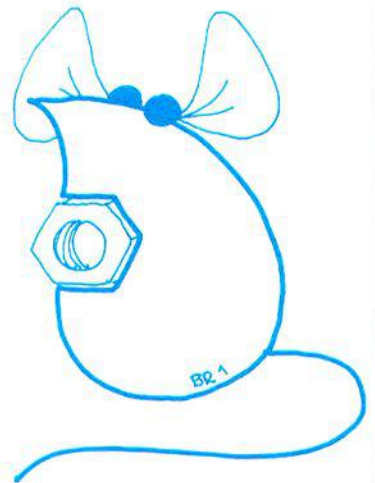
Molto importante, inoltre, è il "tipo" di variabili passate: non è assolutamente concesso di passare stringhe dove sono richiesti numeri interi, ma neanche passare variabil in floating point (virgola mobile) per interi e così via. Questo vale anche per le costanti: se abbiamo dichiarato le variabili riceventi come f.p. (quindi quelle di default) dovremo passare costanti seguite dal punto esclamativo (!) che denota, appunto, grandezze in floating point.

La parola chiave STATIC indica che le variabili del sottoprogramma sono trattate "staticamente", cioè conservano il loro valore all'esterno del programma pur senza influenzare le eventuali omonime. Potremo accedere nuovamente al loro contenuto solo dopo essere rientrati nel sottoprogramma.

L'Enciclopedia di routine in AmigaBasic ha preso inizio sul n. 58 di Commodore Computer Club e si prefigge di offrire periodicamente numerosi sottoprogrammi di utilità generale.

Coloro che intendono contribuire alla divulgazione di routine di un certo interesse, possono telefonarci per proporne la pubblicazione.

```
REM DIMOSTRATIVO USO FUNZIONE SETCOLOR
CALL Setcolor(2, "Nero"): Setcolor 3, "verde"
CLS: COLOR 3, 2: PRINT "Questo e  verde su nero"
Setcolor 0, "ROSSOFUOCO": Setcolor 1, "GIALLO"
COLOR 1, 0: PRINT"giallo su rosso"
FOR j%=1 TO 15000: NEXT: CLS
Setcolor 0, "standard": PRINT "colori Standard"
PRINT "0> BLU, 1> BIANCO, 2> NERO, 3>ARANCIO"
END

' *****************************************
' * Funzione  Setcolor(a, x$) per AMIGABASIC *
' * Assegna al registro 'a', parametro dei    *
' * successivi comandi COLOR, dei valori di   *
' * paletta secondo la stringa definita come *
' * parametro x$. Si possono inserire dei     *
' * nuovi colori col formato: nome, r, g, b   *
' * dove 'NOME' e' del colore in maiuscolo e *
' * r g b sono i tre parametri di PALETTE     *
' *****************************************


SUB Setcolor(x%, x$) STATIC
  x$=UCASE$(x$): RESTORE Setcoldata
  IF x$="STANDARD" THEN
    PALETTE 0, .4, .4, 1: PALETTE 1, 1, 1, 1
    PALETTE 2, 0, 0, 0: PALETTE 3, 1, .73, 0
    COLOR 1, 0: EXIT SUB
  END IF
  FOR j% = 1 TO 14: READ a$, b, c, d
  IF x$=a$ THEN PALETTE x%, b, c, d: EXIT SUB
  NEXT j%
Setcoldata:
DATA NERO, 0.0, 0.0, 0.0, BLUSCURO, 0.4, 0.6, 1.0
DATA MARRONE, .8, .6, .53, GIALLO, 1.0, 1.0, .13
DATA VIOLA, 1.0, .13, .93, ROSSO, 1.0, 0.6, .67
DATA ROSSOFUOCO, .93, .2, 0.0, ARANCIO, 1.0, .73, 0.0
DATA VERDECHIARO, .73, 1.0, 0.0, VERDE, .33, .87, 0.0
DATA BIANCO, 1.0, 1.0, 1.0, GRIGIO, .73, .73, .73
DATA BLUCHIARO, 0.47, 0.87, 1.0, ROSA, .8, 0.0, .93
END SUB
```

```
REM PROGRAMMA ESEMPLIFICATIVO DISPLAYLED - AMIGABASIC
REM Scritto da Luigi R. Callegari per Commodore C. Club
x%=100: y%=30: n$="1234567890": DisplayLED n$, x%, y%, 16
n$="012345": CALL DisplayLED(n$, 90, 80, 30)
n$="678901234567890": DisplayLED n$, 150, 160, 4
END

REM ********* FUNZIONE DI LIBRERIA 'DISPLAYLED' PER AMIGABASIC *************
REM * Mostra una stringa di numeri in caratteri digitali grandi a piacere  *
REM * DisplayLED n$, x%, y%, g%    richiede in INPUT i seguenti parametri: *
REM * n$=stringa di cifre ** x%, y% =posizione sul video ** g% = grandezza *

SUB DisplayLED(n$, x%, y%, g%) STATIC
y0=y% : FOR j%=1 TO LEN(n$) : n=VAL(MID$(n$, j%, 1)) : x0=x%+(j%-1)*2*g%
IF n=0 THEN GOSUB S1: GOSUB S2: GOSUB S3: GOSUB S4: GOSUB S5: GOSUB S6
IF n=1 THEN GOSUB S1: GOSUB S2
IF n=2 THEN GOSUB S6: GOSUB S1: GOSUB S7: GOSUB S4: GOSUB S3
IF n=3 THEN GOSUB S6: GOSUB S1: GOSUB S7: GOSUB S2: GOSUB S3
IF n=4 THEN GOSUB S5: GOSUB S7: GOSUB S1: GOSUB S2
IF n=5 THEN GOSUB S6: GOSUB S5: GOSUB S7: GOSUB S2: GOSUB S3
IF n=6 THEN GOSUB S6: GOSUB S5: GOSUB S7: GOSUB S4: GOSUB S3: GOSUB S2
IF n=7 THEN GOSUB S6: GOSUB S1: GOSUB S2
IF n=8 THEN GOSUB S1: GOSUB S2: GOSUB S3: GOSUB S4: GOSUB S5: GOSUB S6: GOSUB S7
IF n=9 THEN GOSUB S6: GOSUB S1: GOSUB S7: GOSUB S5: GOSUB S2: GOSUB S3
NEXT j%: EXIT SUB
S1: LINE (x0+g%, y0) - STEP (2, g%+2), 3, bf: RETURN
S2: LINE (x0+g%, y0+g%) - STEP (2, g%+2), 3, bf: RETURN
S3: LINE (x0, y0+g%+g%) - STEP (g%, 2), 3, bf: RETURN
S4: LINE (x0, y0+g%) - STEP (2, g%+2), 3, bf: RETURN
S5: LINE (x0, y0) - STEP (2, g%+2), 3, bf: RETURN
S6: LINE (x0, y0) - STEP (g%, 2), 3, bf: RETURN
S7: LINE (x0, y0+g%) - STEP (g%, 2), 3, bf: RETURN
END SUB
```

QUANTO COSTA IL TUO COMMODORE

Amiga 2000 - L. 2.715.000

Microprocessore Motorola MC68000 - Clock 7.16MHz - Kickstart ROM - Memoria RAM: 1 MByte - 3 chip custom per DMA, Video, Audio, I/O - 5 Slot di Espansione Amiga Bus 100 pin Autoconfig™ - 1 Slot di Espansione 86 pin per Schede Coprocessore - 2 Slot di Espansione compatibili AT/XT - 2 Slot di Espansione compatibili XT - 2 Slot di Espansione Video - 1 Floppy Disk Drive da 3 1/2", 880 KBytes - Porta seriale RS232C - Sistema Operativo single-user, multitasking AmigaDOS - Compatibilità MS-DOS XT/AT disponibile con schede interne Janus (A2088 - A2286) - Monitor escluso

Amiga 500 - L. 995.000

Microprocessore Motorola MC68000 - Clock 7.16 MHz - Kickstart ROM - Memoria RAM: 512 KBytes - 3 Chip custom per DMA, Video, Audio, I/O - 1 Floppy Disk Driver da 3 1/2", 880 KBytes - Porta seriale RS232C - Porta parallela Centronics

Videomaster 2995 - L. 1.200.000

Desk Top Video - Sistema per elaborazini video semiprofessionale composto da genlock, digitalizzatore e alloggiamento per 3 drive A2010 - Ingressi videocomposito (2), RGB - Uscite Videocomposito, RF, RGB - sync -

Floppy Disk Driver A 1010 - L. 335.000

Floppy Disk Driver - Drive esterno da 3 1/2" - Capacità 880 KBytes - Collegabile a tutti i modelli della linea Amiga, alla scheda A2088 e al PC1

Floppy Disk Drive A 2010 - L. 280.000

Floppy Disk Drive - Drive interno aggiuntivo da 3 1/2" - Capacità 880 KBytes - Collegabile ad Amiga 2000

Hard Disk A 590 - L. 1.750.000

Hard Disk+Controller+RAM - Scheda Controller - Hard Disk da 3 1/2" 20 MBytes - 2 MBytes "fast" RAM - Collegabile all'Amiga 500

Scheda Janus A 2088 + A 2020 - L. 1.050.000

Scheda Janus XT + Floppy Disk Drive da 5 1/4", 360 KBytes - Scheda Bridgeboard per compatibilità MS-DOS (XT) in Amiga 2000 - Microprocessore Intel 8088 - Coprocessore matematico opzionale Intel 8087

A2286+A2020 - L. 1.985.000

Scheda Janus AT + Floppy Disk Drive da 5 1/4", 1.2 MBytes - Scheda Bridgeboard per compatibilità MS-DOS (AT) in Amiga 2000 - Microprocessore Intel 80287 - Clock 8 MHz - RAM: 1 MBytes on-board - Floppy Disk Controller on-board - Floppy Disk Driver disegnato per l'installazione all'interno dell'Amiga 2000 -

Scheda A2620 - L. 2.700.000

Scheda Processore Alternativo 32 bit - Scheda per 68020 e Unix - Microprocessore Motorola MC68020 - Coprocessore matematico Motorola MC68881 (opzionale MC68882)

Scheda A Unix - L. 3.250.000

Sistema Operativo AT&T Unix System V Release 3 - Per Amiga 2000 con scheda A2620 e Hard Disk 100 MBytes

Hard Disk A2092+PC5060 - L. 1.020.000

Hard Disk e controller - Hard Disk 3 1/2" ST506 - Capacità formattata 20 MBytes

Hard Disk A2090+2092 - L. 1.240.000

Hard Disk e controller - Hard Disk 3 1/2" ST506 - Capacità formattata 20 MBytes

Hard Disk A2090+A2094 - L. 1.900.000

Stesse caratteristiche del kit A2092 ma con disco da 40 MBytes

Espansione di memoria A2058 - L. 1.149.000

Espansione di memoria - Scheda di espansione per Amiga 2000 - Fornita con 2 MBytes "fast" RAM, espandibile a 4 o 8 MBytes

Scheda Video A2060 - L. 165.000

Modulatore video - Scheda modulatore video interna per Amiga 2000 - Uscite colore e monocromatica - Si inserisce nello slot video dell'Amiga 2000

Genlock Card A2301 - L. 420.000

Genlock - Scheda Genlock semiprofessionale per Amiga 2000 - Permette di miscelare immagini provenienti da una sorgente esterna con immagini provenienti dal computer

Professional Video Adapter Card A2351 - L. 1.500.000

Professional Video Adapter - Scheda Video Professionale per Amiga 2000 (B) - Genlock qualità Broadcast - Frame Grabber - Digitalizzatore - Include software di controllo per la gestione interattiva
(Disponibile da maggio '89)

A501 - L. 300.000

Espansione di memoria - Cartuccia di espansione di memoria da 512 KBytes per A500

A520 - L. 45.000

Modulatore RF - Modulatore esterno A500 - Permette di connettere qualsiasi televisore B/N o colori ad Amiga 500

A Scart - L. 28.000

Cavo di collegamento A500/A2000 con connettore per televisione SCART

Monitor a colori 1084 - L. 615.000

Monitor a colori ad alta risoluzione - Tubo 14" Black Matrix antiriflesso - Pitch 0.39 mm - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor a colori 2080 - L. 770.000

Monitor a colori ad alta risoluzione e lunga persistenza - Tubo 14" Black Matrix antiriflesso - Pitch 0.39 mm - Frequenza di raster 50 Hz - Compatibile con Amiga 500/2000, PC (tutta la gamma), C64 e C128

Monitor Monocromatico A2024 - L. 1.235.000

Monitor monocromatico a fosfori "bianco-carta" - Turbo 14" antiriflesso - (Disponibile da marzo '89)

PC60/40 - L. 8.930.000

Microprocessore Intel 80386 - Coprocessore matematico opzionale Intel 80387 - Clock 8 o 16 MHz selezionabile via software e da tastiera - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 funzioni - Sistema Operativo MS-DOS 3 2.1 - Interprete GW-Basic

PC60/40C - L. 9.180.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC 60/80 - L. 10.450.000

Microprocessore Intel 80386 - Coprocessore opzionale Intel 80387 - Clock 8 o 16 MHz selezionabile via software e da tastiera - Memoria RAM 2.5 MBytes - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Floppy Disk Drive opzionale da 3 1/2", 1.44 MBytes - 1 Hard Disk da 80 MBytes - 2 Porte parallele Centronics - Mouse video EGA (compatibile MDA - Hercules - CGA). Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Ambiente Operativo Microsoft Windows/386 - Interprete GW-Basic

PC60/80C - L. 10.700.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC40/20 - L. 4.100.000

Microprocessore Intel 80286 - Coprocessore matematico opzionale Intel 80287 - Clock 6 o 10 MHz selezionabile via software, hardware o da tastiera - Memoria RAM: 1 MByte - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232 - Porta parallela Centronics - Scheda video AGA multistandard (MDA - Hercules - CGA) Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC40/20C - L. 4.350.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC 40/40 - L. 5.285.000

Microprocessore Intel 80286 - Coprocessore matematico opzionale Intel 80287 - Clock 6 o 10 MHz selezionabile via software, hardware o da tastiera - Memoria RAM: 1 MByte - 1 Floppy Disk Drive da 5 1/4", 1.2 MBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232 - Porta parallela Centronics - Scheda video AGA multistandard (MDA - Hercules - CGA) Emulazioni disponibili via hardware e software - Monitor monocromatico 14" - Tastiera avanzata 102 tasti con 12 tasti funzione - Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC40/40C - L. 5.535.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

1352 - L. 78.000

Mouse - Collegabile con Microsoft Bus Mouse - Collegabile direttamente a PC1, PC10/20 - III, PC40 - III

PC910 - L. 355.000

Floppy Disk Drive - Drive interno aggiuntivo da 3 1/2" per PC10/20-I-II-III - Capacità 360 o 720 KBytes selezionabile tramite "config. sys" - Corredo di telaio di supporto per l'installazione in un alloggiamento per un drive da 5 1/4" - Interfaccia identica ai modelli da 5 1/4"

PC1 - L. 995.000

Microprocessore Intel 8088 - 1 Floppy Disk Drive da 5 1/4" - Porta seriale RS232C - Porta parallela Centronics - - Monitor monocromatico 12" - Tastiera 84 tasti - Sistema Operativo MS-DOS 3.2 - Interprete GW-Basic

PCEXP1 - L. 640.000

PC Expansion Box - Box esterno di espansione per PC 1 - Alimentatore aggiuntivo incluso - Contiene 3 Slot di Espansione compatibili Ibm XT - Alloggiamento per Hard Disk da 5 1/4" - Si posiziona sotto il corpo del PC1 e viene collegato tramite degli appositi connettori

PC10-III - L. 1.965.000

Microprocessore Intel 8088 Clock 4.77 MHz 9.54 MHz (double) selezionabile via software e da tastiera - Memoria RAM: 640 KBytes - 2 Floppy Disk Drive da 5 1/4", 360 KBytes - Porta seriale RS232C - Porta parallela Centronics - Porta Mouse per Mouse Commodore 1352 (compatibile Microsoft Bus Mouse - Tastiera avanzata 102 con 12 tasti funzione Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC10-IIIC - L. 2.300.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

PC20-III - L. 2.715.000

Microprocessore Intel 8088 - Clock 4.77 MHz 9.54 MHz (double) selezionabile via software e da tastiera - 1/4", 360 KBytes - 1 Hard Disk da 20 MBytes - Porta seriale RS232C - Porta parallela Centronics - Porta Mouse per Mouse Commodore 1352 (compatibile Microsoft Bus Mouse) - Tastiera avanzata 102 con 12 tasti funzione Sistema Operativo MS-DOS 3.21 - Interprete GW-Basic

PC20-IIIC - L. 3.050.000

Stessa configurazione ma con monitor 14" a colori mod. 1084

Nuovo C64 - L. 325.000

Nuovo Personal Computer CPU 64 KBytes RAM - Vastissima biblioteca software disponibile - Porta seriale Commodore - Porta registratore per cassette - Porta parallela programmabile -

C128D - L. 895.000

Personal Computer CPU 128 KBytes RAM espandibile a 512 KBytes - ROM 48 KBytes - Basic 7.0 - Tastiera separata - Funzionante in modo 128.64 o CP/M 3.0 - Include floppy disk drive da 340 KBytes

Floppy Disk Drive 1541 II - L. 365.000

Floppy Disk Drive - Floppy Disk Drive da 5 1/4" singola faccia - Capacità 170 KBytes - Alimentazione separata - Compatibile con C64, C128, C128D

Floppy Disk Dirve 1581 - L. 420.000

Floppy Disk Drive da 3 1/2" doppia faccia - Capacità 800 KBytes - Alimentazione separata - Compatibile con C64, C128, C128D

1530 - L. 55.000

Registratore a cassette per C64, C128, C128D

Accessori per C64 - 128D

1700 - Espansione di memoria - Cartuccia di espansione di memoria a 128 KBytes per C128 **- L. 170.000**

1750 - Espansione di memoria - Cartuccia di espansione di memoria 512 KBytes per C128 - **L. 245.000**

1764 - Espansione di memoria - Cartuccia di espansione di memoria a 256 KBytes per C64
Fornita di alimentatore surdimensionato - **L. 198.000**

16499 - Adattatore Telematico Omologato - Collegabile al C64
Permette il collegamento a Videotel, P.G.E. e banche dati - **L. 149.000**

1399 - Joystick - Joystick a microswitch con autofire - **L. 29.000**

1351 - Mouse - Mouse per C64, C128, C128D - **L. 72.000**

Monitor Monocromatico 1402 - L. 280.000

Monitor monocromatico a fosfori "bianco-carta" - Turbo 12" antiriflesso - Ingresso TTL - Compatibile con tutta la gamma PC

Monitor Monocromatico 1404 - L. 365.000

Monitor monocromatico a fosfori ambra - Turbo 14" antiriflesso a schermo piatto - Ingresso TTL - Compatibile con tutta la gamma PC - Base orientabile

Monitor Monocromatico 1450 - L. 470.000

Monitor monocromatico BI-SYNC a fosfori "bianco-carta" - Turbo 14" antiriflesso - Ingresso analogico e digitale - Doppia frequenza di sincronismo orizzontale per compatibilità con adattatori video MDA, Hercules, CGA, EGA e VGA

Monitor a colori 1802 - L. 445.000

Monitor a colori - Turbo 14" - Collegabile a C64, C128, C128D

Monitor monocromatico 1900 - L. 199.000

Monitor monocromatico a fosfori verdi - Turbo 12" antiriflesso - Ingresso videocomposito - Compatibile con tutta la gamma Commodore

Monitor a colori 1950 - L. 1.280.000

Monitor a colori BI-SYNC alta risoluzione - Turbo 14" antiriflesso - Ingresso analogico e digitale - Doppia frequenza di sincronismo orizzontale per compatibilità con adattatori video MDA, Hercules, CGA, EGA e VGA

Stampante MPS 1230 - L. 465.000

Stampante a matrice di punti - Testina a 9 aghi - 120 cps - Bidirezionale - 80 colonne - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia seriale Commodore e parallela Centronics - Compatibile con tutti i prodotti Commodore

MPS 1230R - L. 19.000

Nastro per stampante

Stampante MPS 1500C - L. 550.000

Stampante a colori a matrice di punti - Testina a 9 aghi - 130 cps - Bidirezionale - 80 colonne - Supporta nastro a colori o nero - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia parallela Centronics - Compatibile con la gamma Amiga e PC

MPS1500R - L. 37.000

Nastro a colori per stampante

Stampante MPS 1550C - L. 575.000

Stampante a colori a matrice di punti - Testina a 9 aghi - 130 cps - Bidirezionale - 80 colonne - Supporta nastro a colori o nero - Near Letter Quality - Stampa grafica - Fogli singoli e modulo continuo - Trascinamento a trattore e/o frizione - Interfaccia seriale Commodore e parallela Centronics - Compatibile con tutti i prodotti Commodore

I COMMODORE POINT

LOMBARDIA

Milano
- AL RISPARMIO - V.le Monza, 204
- BCS - Via Montegani, 11
- BRAHA ALBERTO - Via Pier Capponi, 5
- E.D.S. - Corso Porta Ticinese, 4
- E.S.C. - Via Roggia Scagna, 7
- FAREF - Via A. Volta, 21
- FLOPPERIA - Viale Monte Nero, 31
- GBC - Via Cantoni, 7 - Via Petrella, 6
- GIGLIONI LAURA - Via D'Ovidio, 8
- GILGLIONI - Viale Luigi Sturzo, 45
- LOGITEK - Via Golgi, 60
- MARCI[illegible] - Via F.lli Bronzetti, 37
- ME[illegible]HIONI - Via P. Colletta, 37
- MESSAGGERIE MUSICALI - Galleria del Corso, 2
- NEWEL - Via Mac Mahon, 75
- RIVOLA - Via Vitruvio, 43

Provincia di Milano
- IL CURSORE - Via Cavour, 35 - Novate Milanese
- REC ELGRA - Corso Milano, 118 - Bovisio Masciago
- F.LLI GALIMBERTI - Via Nazionale dei Giovi, 28/36 - Barlassina
- GBC - Viale Matteotti, 66 - Cinisello Balsamo
- P. GIORGIO OSTELLARI - Via Milano, 300 - Desio
- CASA DELLA MUSICA - Via Indipendenza, 21 - Cologno Monzese
- PENATI - Via Verdi, 28/30 - Corbetta
- EMP - Viale Italia, 12 - Corsico
- CENTRO COMPUTER PANDOLFI - Via Corridoni, 18 - Legnano
- COMPUTEAM - Via Vecellio, 41 - Lissone
- FUTURA - Via Solferino, 31 - Lodi
- M.B.M. - Corso Roma, 12 - Lodi
- L'AMICO DEL COMPUTER - Viale Lombardia, 17 - Melegnano
- BIT 84 - Via Italia, 4 - Monza
- I.C.O. - Via dei Tigli, 14 - Opera
- NIWA HARD & SOFT - Via Bruno Buozzi, 94 - Sesto S. Giovanni
- FERRARI LUIGI - Via Madre Cabrini - Sant'Angelo Lodigiano
- IL COMPUTER SERVICE SHOP - Via Padana Superiore, 197 - Vimodrone

Bergamo
- COMIF - Via Autolinee, 10
- CORDANI - Via dei Caniana, 8
- D.R.B. - Via Borgo Palazzo, 65
- NEW SYSTEMS - Via Paglia, 36

Provincia di Bergamo
- BERTULEZZI G. - Via Fantoni, 48 - Alzano Lombardo
- COMPUTER TEAM - Via Verdi, 1/B - Carvico
- OTTICO ROVETTA - Piazza Garibaldi, 6 - Lovere
- A.I.S. INTERNATIONAL - Via San Carlo, 25

Aan Pellegrino Terme
- SISTHEMA - Via Roma, 45 - Sarnico
- COMPUTER POINT - Via Lantieri, 52 - Sarnico
- AB INFORMATICA - Statale Cremasca, 66 - Urgnano

Brescia
- COMPUTER CENTER - Via Cipro, 6
- INFORMATICA 2000 - Via Stazione, 16/B
- MASTER INFORMATICA - Via F.lli Ugoni, 10/B
- VIGASIO MARIO - Port. Zanardelli, 3

Provincia di Brescia
- MISTER BIT - Via Mazzini, 70 - Breno
- CAVALLI PIETRO - Via 10 Giornate, 14/B - Castrezzato
- VIETTI GIUSEPPE - Via Milano, 1/B - Chiari
- MEGABYTE - Piazza Malvezzi, 14 - Desenzano del Garda
- CO-RE - Via XXV Aprile, 136 - Fiero
- BARESI RINO & C. - Via XX Settembre, 7 - Ghedi
- INFO CAM - Via Provinciale, 38 - Gratacasolo

Como
- IL COMPUTER - Via Indipendenza, 90
- 2M ELETTRONICA - Via Sacco, 3

Provincia di Como
- ELTRONGROS - Via L. da Vinci, 54 - Barzanò
- EGA - Via Mazzini, 42 - Cassago Brianza
- DATA FOUND - Via A. Volta, 4 - Erba
- EGA - Via A. Moro, 17 - Galbiate
- RIGHI ELETTRONICA - Via G. Leopardi, 26 - Olgiate Comasco
- CIMA ELETTRONICA - Via Leonardo da Vinci, 7 - Lecco
- FUMAGALLI - Via Cairoli, 48 - Lecco

Cremona
- MONDO COMPUTER - Via Giuseppina, 11/B
- PRISMA - Via Buoso sa Dovara, 8
- TELCO - Piazza Marconi, 2/A

Provincia di Cremona
- ELCOM - Via IV Novembre, 56/58 - Crema
- EUROELETTRONICA - Via XX Settembre, 92/A - Crema

Mantova
- TUBALDO ELSO & C. - Galleria Fermi, 7
- 32 BIT - Via Cesare Battisti, 14
- ELETTRONICA BASSO - V.le Risorgimento, 69

Pavia
- POLIWARE - C.so C. Alberto, 76

Provincia di Pavia
- LOGICA INFORMATICA - Via Monte grappa, 32 - Vigevano
- M. VISENTIN - C.so V. Emanuele, 76 - Vigevano

Sondrio
- CIPOLLA M. - Via Tremogge, 25

Provincia di Sondrio
- FOTONOVA - Via Valeriana, 1 - San Pietro di Berbenno

Varese
- DIMECO SISTEMI - Via Garibaldi
- IL CENTRO ELETTRONICO - Via Morazzone, 2
- SUPERGAMES - Via Carrobbio, 13

Provincia di Varese
- BUSTO BIT - Via Gavinana, 17 - Busto Arsizio
- CRESPI G.&C. - V.le Lombardia, 59 - Castellanza
- COMPUTER SHOP - Via A. da Brescia, 2 - Gallarate
- LIMA IMPORT-EXPORT - (Grandi Magazzini Bossi) - Via Clerici, 196 - Gerenzano
- J.A.C. - Via Matteotti, 38 - Sesto Calende

VALLE D'AOSTA

Aosta
- F.lli GATTI - Via Festaz, 75

PIEMONTE

Alessandria
- BIT MICRO - Via Mazzini, 104
- SERVIZI INFORMATICI - Via Alessandro III, 47

Provincia di Alessandria
- S.G.E. ELETTRONICA - Via Bandello, 19 - Tortona

Asti
- RECORD - Corso Alfieri, 166/3

Cuneo
- ROSSI COMPUTER - C.so Nizza, 42
- STUDIO Software - C.so Nizza, 49

Provincia di Cuneo
- PUNTO BIT - Corso Langhe, 26/C - Alba
- SDI - Via Vittorio Emanuele, 250 - Bra
- ASCHIERI GIANFRANCO - Corso Emanuele Filiberto, 6 - Fossano

Novara
- ELCOM - Corso Mazzini, 11
- PROGRAMMA 3 - V.le Buonarroti, 8
- PUNTO VIDEO - Corso Risorgimento, 39/B

Provincia di Novara
- MIRCO POLACCO & C. - Via Monte Zeda, 4 - Arona
- ALL COMPUTER - Corso Garibaldi, 106 - Borgomanero
- MICROLOGIC - Via Giovanni XIII, 2 - Domodossola
- ELLIOT COMPUTER - Via Don Minzoni, 32 - Intra

Torino
- ALBA ELETTRONICA - Via C. Fossati, 5/P
- ALEX COMPUTER - Corso Francia, 333/4
- COMPUTER HOME - Via San Donato, 46/D
- COMPUTING NEWS - Via Marco Polo, 40/E
- DE BUG - C.so V. Emanuele II, 22
- DESME UNIVERSAL - Via San Secondo, 95
- F.D.S. - Via Borgaro, 86/D
- INFORMATICA ITALIA - Corso Re Umberto, 128
- MT INFORMATICA - Corso Giulio Cesare, 58
- MUSICS'S Shop - C.so Potenza, 177
- NEW BUSINESS COMPUTER - Via Nizza, 45/F
- PLAY GAMES - Via C. Alberto, 39/A
- RADIO TV MIRAFIORI - C.so Unione Sovietica, 381
- SMIT ELETTRONICA - Via Bibiana, 83/B
- TELERITZ - Corso Traiano, 34

Provincia di Torino
- PAUL E CHICO VIDEOSOUND - Via V. Emanuele, 52 - Chieri
- BIT INFORMATICA - Via V. Emanuele, 154 - Ciriè
- HI-FI CLUB - Corso Francia, 92/C - Collegno
- I.C.S. - Stradale Torino N. 73 - Ivrea
- BAS - Corso Roma, 47 - Moncalier
- CERUTTI MAURO - Corso Torino, 234 - Pinerolo
- EUREX Corso Indipendenza, 5 - Rivarolo C.se
- DIAM INFORMATICA - C.so Francia, 146/Bis - Rivoli
- FULLINFORMATICA - Via Vittorio Veneto, 25 - Rivoli
- GAMMA COMPUTER - Via Cavour, 3 A/B - Settimo Torinese

Vercelli
- ELETTROGAMMA - C.so Bormida, 27
- ELETTRONICA di BELLAMO A. & C. - Strada Torino, 15

Provincia di Vercelli
- C.S.I. TEOREMA - Via Losana, 9 - Biella
- SIGEST - Via Bertodano, 8 - Biella
- REMONDINO Franco - Via Roma, 5 - Borgosesia
- FOTOSTUDIO TREVISAN - Via XXV Aprile, 24/B - Cossato
- STUDIO FOTOGRAFICO IMARISIO - Piazza Martiri Libertà, 7 - Trino

LIGURIA

Genova

ABM COMPUTER - Piazza De Ferrari, 24/R
- CENTRO ELETTRONICA - Via Chiaravagna, 10/R - Località Sestri Ponente
- COMMERCIALE SOTTORIPA - Via Sottoripa, 115/117
- FOTOMONDIAL - Via del Campo, 3-5-9-11-13/R
- LA NASCENTE - Via San Luca, 4/1
- RAPPR - EL - Via Boreoratti, 23/R

Imperia
- CASTELLINO - Via Belgrano, 44
- SASA COMPUTER - Via Nazionale, 256

Provincia di Imperia
- CENTRO HI-FI VIDEO - Via della Repubblica, 38 - Sanremo
- CASTELLINO - Via Genova, 48 - Ventimiglia

La Spezia
- CATTONI - Via Vitt. Veneto, 75
- I.L. ELETTRONICA - Via Vitt. Veneto, 123

Provincia di La Spezia
- I.L. ELETTRONICA - Via Aurelia, 299 - Fornola di Vezzano

Savona
- CASTELLINO - Corso Tardy e Benech, 101
- ATHENA - Via Carissimo E. Crotti, 16/R

VENETO

Belluno
- UP TO DATE - Via Vittorio Veneto, 43

Provincia di Belluno
- GUERRA COMPUTERS - Viale Mazzini, 10/C - Feltre

Padova
- BIT SHOP - Via Cairoli, 11
- COMPUMANIA
- Riviera Tiso Camposanpiero, 37
- COMPUTER POINT - Via Roma, 63
- D.P.R. - V.le Lombardo, 4
- GIANFRANCO MARCATO
- Via Madonna della Salute, 51/53
- ZELLA ADELIO
- Piazza De Gasperi, 31/A

Provincia di Padova
- CAERT - Via Andorra, 11
- Zona Industriale Camin

Treviso
- BIT 2000 - Via Brandolini d'Adda, 14

Provincia di Treviso
- DE MARIN - Vai XX Settembre, 74
- Conegliano
- SIDESTREET - Via S. d'Acquisto, 8
- Montebelluna
- FALCON - Via Terraggio, 116
- Preganziol

Venezia
- TELERADIO FUGA
- San Marco, 3457

Provincia di Venezia
- GUERRA EGIDIO & C.
- Via Bissuola, 20/A - Mestre
- TREKILOWATT
- Via Torre Belfredo, 47 - Mestre
- REBEL - Via F. Crispi, 10
- San Donà di Piave
- GUERRA COMPUTERS
- Via Vizzotto, 29
- San Donà di Piave
- TELFERT - Via Chiesa, 1509
- Sottomarina
- RADIOCESTARO - Via Roma, 89
- Spinea

Verona
- CASA DELLA RADIO - Via Cairoli, 10
- TELESAT - Via Vasco de Gama, 8

Provincia di Verona
- FERRARIN - Via dei Massari, 10
- Legnago

Vicenza
- ELETTRONICA BISELLO
- Viale Trieste, 427/429
- SCLACHI Market - Via Cà Balbi, 139

Provincia di Vicenza
- GUERRA COMPUTERS
- Via Dell'Industria - Alte Ceccato
- SCIAVOTTO - Via Zanella, 21
- Cavazzale
- COMPUTER B. COSTO
- Via del Costo, 34 - Thiene
- ELETTROCASA - Via Roma, 67
- Trissino

FRIULI VENEZIA GIULIA

Gorizia
- E.C.O. ELETTRONICA
- Via F.lli Cossar, 23

Pordenone
- RIGO - Viale Cossetti, 5

Provincia di Pordenone
- MDT - Piazza Repubblica, 5
- Villanova di Prata
- BRUNO DA PIEVE
- Via Colombera, 17 - Porcia

Trieste
- AVANZO GIACOMO
- Piazza Cavana, 7
- COMPUTER SHOP - Via P. Reti, 6
- COMPUTIGI - Via XX Settembre, 51
- CTI - Via Pascoli, 4

Udine
- MOFERT 2 - Via Leopardi, 21
- R.T. SYSTEM - Via L. da Vinci, 99

Provincia di Udine
- IDRENO MATTIUSSI & C.
- Via Liciniana, 50 - Tavagnacco

Rovigo
- CLINICA DEL RASOIO
E DEL COMPUTER
- Via Fiume, 31/33

TRENTINO ALTO ADIGE

Bolzano
- C.M.B. ITALIA - Via Roma, 82
- MATTEUCCI PRESTIGE
- Via Museo, 54

Provincia di Bolzano
- ELECTRO TAPPEINER
- P.za Principale, 90 - Silandro
- RADIO MAIR - Via Centrale, 70
- Brunico
- ELECTRO RADIO HENDRICH
- Via delle Corse, 106 - Merano

Trento
- CRONST - Via Galilei, 25

EMILIA ROMAGNA

Piacenza
- COMPUTER Line - Via G. Carducci, 4
- DELTA COMPUTER
- Via Martiri della Resistenza, 15/G
- SOVER - Via IV Novembre, 60

TOSCANA

Arezzo
- DELTA SYSTEM - Via Piave, 13

Firenze
- ATEMA
- Via Bendetto Marcello, 1a/1b
- COOPERATIVA L.D.T. - Via Icaro, 9
- ELETTRONICA CENTOSTELLE
- Via Centostelle, 5/a
- HELP COMPUTER
- Via degli Artisti, 15/A
- PUNTO SOFT - Via Vagnetti, 17
- TELEINFORMATICA TOSCANA
- Via Bronzino, 36

Provincia di Firenze
- WAR GAMES
- Via Raffaello Sanzio, 126/A
- Empoli
- NEW E.V.M. COMPUTER
- Via degli Innocenti, 2
- Figline Valdarno
- COSCI F.LLI - Via Roma, 26 - Prato
- CENTRO INFORMATICA
- Via F.lli Cervi, 21/29 - Pontassieve

Grosseto
- COMPUTER SERVICE
- Piazza Ponchielli, 2

Livorno
- ETA BETA - Via San Francesco, 30
- FUTURA 2 - Via Cambini, 19

Provincia di Livorno
- ELETTRONICA ALESSI PAOLO
- Via Cimarosa, 1 - Piombino

Provincia di Lucca
- IL COMPUTER - V.le Colombo, 216
- Lido di Camaiore
- SANTI VITTORIO - Via Roma, 23
- San Romano Garfagnana

Massa
- EURO COMPUTER
- Piazza Bertagnini, 4
- FIRMWARE - Via Aurelia Ovest, 27

Provincia di Massa
- RADIO LUCONI - Via Roma, 24/B
- Carrara

Pisa
- C.H.S. - Via Carlo Cattaneo, 90/92
- ELECTRONIC SERVICE
- Via della Vecchia Tranvia, 10
- IT - LAB - Via Marche 8A/8B

Pistoia
- ELECTRONIC SHOP
- Via della Madonna, 49
- OFFICE DATA SERVICE
- Galleria Nazionale, 22

Provincia di Pistoia
- ZANNI & C. - Corso Roma, 45
- Montecatini Terme

Siena
- VIDEO MOVIE - Via Garibaldi, 17

Provincia di Siena
- ELECTRONIC Shop - Via A. Casini, 51
- Chianciano Terme
- ELETTRONICA
- Via di Gracciano nel Corso, 111
- Montepulciano

UMBRIA

Provincia di Perugia
- COMPUTER STUDIOS
- Via IV Novembre, 18/A
- Bastia Umbra

CAMPANIA

Provincia di Avellino
- FLIP FLOP - Via Appia, 68 -Atripalda

Benevento
- E.CO: INFORMATICA
- Via Pepicelli, 21/25

Caserta
- O.P.C. - Via G.M. Bosco, 24

Provincia di Caserta
- M.P. COMPUTER - Via Napoli, 30
- Maddaloni

Napoli
- BABY TOYS
- Via Cisterna dell'Olio, 5/Bis
- CASA MUSICALE RUGGIERO
- Piazza Garibaldi, 74
- CENTRO ELETTRONICO CAMPANO
- Via Epomeo, 121
- C.I.AN - Galleria Vanvitelli, 32
- DARVIN - Calata San Marco, 26
- ELETTRONICA RO.DA.LO.
- Via Epomeo, 216/B
- GIANCAR 2 - Piazza Garibaldi, 37
- GRUPPO BUSH
- Galleria Umberto I, 55
- ODORINO - Largo Lala, 22/A-B
- R 2 - Via F. Cilea, 285
- SPY - Via Fontana, 135
- TOP - Via S. Anna dei Lombardi, 12
- VIDEOFOTOMARKET
- Via S. Brigida, 19

Provincia di Napoli
- SPADARO - Via Romani, 93
- S. Anastasia
- TUFANO - S.S. Sannitica, 87 Km 7
- Casoria
- ELETTRONICA 2000
- Corso Durante, 40
- Frattamaggiore
- GATEWAY - Via Napoli, 68
- Mugnano
- NUOVA INFORMATICA SHOP
- Via Libertà, 185/191 - Portici
- BASIC COMPUTER
- C.so Garibaldi, 34
- Pozzuoli
- FALCO ELETTRONICA
- Via Sarno, 100 - Striano
- TECNOTRE - Via P. Fusco, 1/F
- Torre Annunziata

Salerno
- COMPUTER MARKET
- C.so Vitt. Emanuele, 23

Provincia di Salerno
- KING COMPUTER - Via Olevano, 56
- Battipaglia
- DIMER POINT - Via C. Rosselli, 20
- Eboli

PUGLIA

Bari
- ARTEL - Via G. d'Orso, 9
- COMPUTER'S ARTS
- Viale Meucci, 12/B

Provincia di Bari
- F. FAGGELLA - Corso Garibaldi, 15
- Barletta
- G. FAGGELLA
- Via P. d'Aragona, 62/A - Barletta
- G. LONUZZO - Via Nizza, 21
- Castellana

Brindisi
- MARANGI & MICCOLI
- Via Prov. San Vito, 165

Provincia di Foggia
- IL DISCOBOLO - Via T. Solis, 15
- San Severo

Lecce
- BIT
- Via 95° Regg.to Fanteria, 87/89

Provincia di Lecce
- CEDOK INFORMATICA - Via Roma, 31
- Tricase

Matera
- GUADIANO ELECTRONICS
- Via Roma, 1

Taranto
- ELETTROJOLLY - Via de Cesare, 13
- TEA - Via Regina Elena, 101

CALABRIA

Catanzaro
- C. & G. COMPUTER - Via F. Acri, 28
- PAONE SAVERIO - Via F. Acri, 93/99

Provincia di Catanzaro
- COMPUTER HOUSE - Via Bologna
- Crotone
- OTTICA FOTO NELLO RUELLO
- C.so Vittorio Emanuele, 177
- Vibo Valentia

Cosenza
- SIRANGELO COMPUTER
- Via N. Parisio, 25

Provincia di Cosenza
- ELIGIO ANNICHIARICO & C.
- Via Roma, 21 - Castrovillari
- ALFA COMPUTER
- Via Nazionale, 341/A
- Corigliano Scalo
- ING. FUSTO SALVATORE
- Corso Nicotera, 99
- Lamezia Terme

Reggio Calabria
- CONTROL SYSTEM
- Via S. Francesco da Paola, 49/DE
- SYSTEM HOUSE
- Via Fiume ang. Palestino, 1

Provincia di Reggio Calabria
- COMPUTER SHOP
- Via Matteotti, 50/52 - Locri

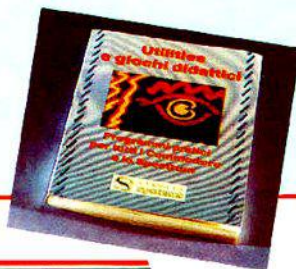
PRODOTTI SYSTEMS EDITORIALE

Software su cassetta

La voce III	L.12000
Raffaello	L.10000
Oroscopo	L.12000
Computer-Music	L.12000
Gestione familiare	L.12000
Banca dati	L.12000
Dichiarazione dei redditi (740/S)	L.16000
Matematica finanziaria	L.20000
Analisi di bilancio	L.20000
Arredare (richiede linguaggio Simon's Basic)	L.10000

Software su disco

Ms-Dos & Gw-Basic	L.25000
La voce III	L.12000
Oroscopo	L.12000
Gestione familiare	L.12000
Banca dati	L.12000
Dichiarazione dei redditi (740/S)	L.24000
Matematica finanziaria	L.20000
Analisi di bilancio	L.20000
Arredare (richiede linguaggio Simon's Basic)	L.20000
Graphic Expander C/128 in modo 80 colonne	L.27000

Offerta "Commodore speciale L.M." + dischetto L.16000

Libri

64 programmi per il Commodore 64	L. 4800
I miei amici C/16 e Plus/4	L. 7000
Strategie vincenti per Commodore 64	L. 5800
62 programmi per Vic 20, C/16 e Plus/4	L. 6500
Tutti i segreti dello Spectrum	L. 7000
Impara giocando il Basic dello Spectrum	L. 7000
Micro Pascal per Commodore 64/128	L. 7000
Dal registratore al drive del C/64	L. 7000
Ada	L. 5000
Il linguaggio Pascal	L. 5000

Directory

Ciascun dischetto	L.12000

Arretrati

Ciascun numero arretrato di Commodore Computer Club	L. 5000
Ciascun numero arretrato di Personal Computer	L. 5000
Ciascun numero arretrato di VR Videoregistrare	L. 5000

Sconti e agevolazioni

Le spese di imballo e spedizione sono a carico della Systems se ciascun ordine è pari ad almeno L.50000 (di listino).

Gli abbonati hanno diritto allo sconto del 10% e alla spedizione gratuita se la somma totale raggiunge la cifra di L.50000 (di listino).

Abbonamenti

Commodore Computer Club (11 fascicoli)	L. 50.000
Personal Computer (11 fascicoli)	L. 45.000
Commodore Computer Club + Personal Computer (11 + 11 fascicoli)	L. 95.000
VR Videoregistrare (12 numeri)	L. 50.000

N.B.: la cifra per gli abbonamenti non può essere conteggiata per ottenere gli sconti e le agevolazioni di cui sopra.

Compilate un normale modulo di C/C postale indirizzando a:

C/C postale N. 37952207
Systems Editoriale
Viale Famagosta, 75
20142 Milano

Non dimenticate di indicare chiaramente, sul retro del modulo (nello spazio indicato con "Causale del versamento") non solo il vostro nominativo completo di recapito telefonico, ma anche il materiale desiderato.

In ogni caso sarebbe opportuno inviare la presente scheda, debitamente compilata, allegando la fotocopia della ricevuta del versamento effettuato.

Chi volesse ricevere più celermente la confezione deve inviare la somma richiesta mediante assegno circolare oppure normale assegno bancario (non trasferibile o barrato due volte) intestato a: Systems Editoriale - Milano.

Per un ottimale utilizzo del software "Linguaggio Macchina e Routine grafiche per C/64" è opportuna la lettura del fascicolo "Commodore Speciale" appositamente dedicato.

Coloro che desiderano procurarsi i prodotti della Systems Editoriale devono inviare, oltre alla cifra risultante dalla somma dei singoli prodotti, la cifra di L.3000 per spese di imballo e spedizione, oppure L.6000 se si preferisce la spedizione per mezzo raccomandata.